A Weighted Grey Relational Model for Classifying Software Quality

Nan-Hsing Chiu
Department of Information Management,
Ching Yun University, Taiwan
E-mail: nhchiu@cyu.edu.tw

A. M. Masuduzzaman
Department of Applied Foreign Language,
Ching Yun University, Taiwan
E-mail: drmasud4@cyu.edu.tw

Abstract-*Grey relational analysis is a well known approach that is utilized for generalizing estimates under small sample and uncertain conditions. This study examines the potential benefits for providing a software quality classification based on weighted grey relational classifier. The particle swarm optimization approach is adopted to explore the best fit of weights on software metrics in the grey relational analysis approach. Empirical results show that the proposed approach provides a preferred balance of misclassification rates than the grey relational classifiers without using PSO. It also outperforms the widely used decision tree approaches.*

Keywords: Software Project Management, Software Quality Classification, Grey Relational Analysis, Particle Swarm Optimization.

1. Introduction

The software development process imposes major impacts by providing an acceptable software quality in accordance with the specification as user required. Prior to the system test, identifying possible faulty software components or software modules can effectively improve the testing efforts. Software quality classification thus aims to evaluate software quality level and to indicate software problems at an early stage. This enables software project managers to effectively allocate the finitely testing resources on high-risk modules. During these decades, various software quality classification techniques has been developed for targeting high-risk modules based on different software metrics [3],[6],[13]. However, most software products have incomplete information and uncertain relations between software metrics and the software quality.

The grey relational analysis (GRA) is an important approach in the grey system theory that has become very popular in many fields [10],[11]. It is essentially believed to have captured the similarity measurements or relations between the case being estimate and historical cases for solving classification problems. However, identifying the significant factors that strongly influence dependent variable is a crucial issue for GRA approach [4]. Particle swarm optimization (PSO) is a search algorithm that has been successfully applied to a large number of difficult optimization problems [17],[14]. The current study aims to examine the potential benefits of software quality classification using the weighted GRA (WGRA) in order to provide a preferred model for project managers.

2. Literature Review

During these decades, a large number of software quality classification models have been proposed in the literature. These approaches include classification and regression trees (CART) [7], C4.5 decision trees [9], and many other methods [3],[6],[13]. A software quality classification model may not be useful to guide software improvement efforts if either type of misclassification rate is high. From a practical point of view, a proper software quality classification model often required a good balance between these two types of misclassifications [8]. In addition to good balance between these two misclassification rates, the characteristics of software metrics also play a crucial role on the predictive performances [9]. One of the most crucial challenges in the classification of software quality is targeting high-risk modules accurately from the vague and incomplete information. It is extremely difficult to accurately identify relevant software metrics that strongly influence software quality because the degree of influence is imprecise in nature [15].

In 1982, Deng initiated the grey system theory that is applicable to study uncertainties in system models, analyze relations between systems, establish models, and make forecasts and decisions [2]. In the grey system theory, the GRA is essentially believed to have captured the similarity measurements or relations between case being estimate and historical cases in order to make forecasts [4]. The grey relational grade (GRG) of GRA shows the degree of relationship between the case being estimated and each case in the historical dataset. For a classification model based on GRA approach, the retrieval of the best comparative case for forecasting relies on a relational measure that takes into account of the GRG between pairs of cases. A grey relational classifier is crucial to identify the most prominent independent variable that leads to the effective case retrieval. Generally, more important independent variables should be assigned larger weights in compared with less important ones in order to determine the significant software metrics for software quality classification.

PSO algorithm is a search method that is an easy implementation for solving optimization problems. In 1995, Kennedy and Eberhart introduced it as a novel evolutionary computation technique with a capability to optimize complex numerical problems [5]. The concept of PSO was motivated from the simulation of birds flocking that mimics the social behavior of flying birds and their means of information exchange with companions to explore the optimized solution. Each potential solution is seen as a particle with a certain velocity and flies through the problem space, and adjusts its flight toward the globally optimal solution according to its own flying experience and its companions' flying experience. This approach explores optimal solution in complex search spaces through the interaction of individuals in a population of particles. It does not need complex operators such as crossovers and mutations in compared with the evolutionary computation technique of genetic algorithms. Instead of genetic algorithms, it only requires simple mathematical operators and computationally inexpensive runtime [16].

The performance of the relational measures and the weights of the each software complexity metric are critical to the reasoning process for GRA. PSO is a useful method that can explore good solutions for solving many complexity problems, especially for discovering weights. However, little research has attempted to adopt the PSO approach to optimize weights in the GRA for software quality classification. Thus, this paper aims to investigate the effects of the software quality classification based on an WGRA approach that integrates the GRA with PSO for providing a practical software quality classification model with a good balance between the *TIE* (Type I error) and *TIIE* (Type II error) rates.

3. Methodology

Applying GRA for software quality classification includes four steps of grey relational generating, grey relational coefficient, GRG and grey relational rank. A WGRA approach for software quality classification that integrates the GRA with PSO is shown in Fig. 1. Grey relational generating is used to replenish messages from the data by finding regularities and properties in jumbled software metrics. It is also aware of data normalization procedures. For each software metric, the grey relational coefficient is calculated to express the relation between the software module being classified and historical software modules. The GRG is a global measure adopted for the GRA. The greater GRG between two software modules, the closer the relationship between these software modules are. All the software modules are ranked in accordance with their GRGs which is called the grey relational rank. For the software module being classified, the classification result should be retrieved from the software module base with the largest GRG among all historical software modules.

For the PSO, it is applied to search for suitable weights in conjunction with the weighted GRG in the WGRA. These weights in the weighted GRG indicate the significance of software metrics in determining the similarity between two software modules. The initial weight stage randomly generates feasible solutions as well as initial weights for WGRA. The fitness evaluation stage makes sure the current classification result whether is achieved or not. The best weight updating stage modifies the current best weights as the personal best (*pbest*) or global best (*gbest*), and the new weight generation stage produces the new weights for weighted GRG. These iterative processes explore the suitable weights which produce a good balance between the *TIE* and *TIIE* rates among a set of software modules being classified.

The PSO is applied to search for the suitable weights in conjunction with the grey relational coefficients for the weighted GRG in the GRA. It involves simulating social behavior among particles flying through a multi-dimensional search space. The weights in the weighted GRG indicate the significance of software metrics in determining the similarity between software module being classified and historical software modules. The weights of weighted GRG are treated as variables to be searched, and a set of candidate weights for a software module is represented as a particle. Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far.

For the weight exploration on GRA via PSO, at the beginning, the PSO generates a population of particles at random which makes it possible to explore a broad population of possible weights in the entire search space. Each particle includes a position vector and a moving velocity which represented as x and v, respectively. For the optimization of m weights in a software module, the position and velocity of the k^{th} software metric in the p^{th} particle is represented as $x_p(k)=\{x_{p,1}, x_{p,2}, \ldots, x_{p,k}\}$ and $v_p(k)=\{v_{p,1}, v_{p,2}, \ldots, v_{p,k}\}$. Each particle keeps track of its weight, which is uniformly spread for each of the software metric dimensions in the solution space. The position of the individual best fitness that the p^{th} particle has achieved so far is known as the personal best which is denoted as $x_p^{best}(k)$. The highest fitness that has been obtained among all the particles in the population so far is known as the global best which is denoted as $x^{best}(k)$. Both the personal best and the global best are stored for generating the new velocity of the p^{th} particle.

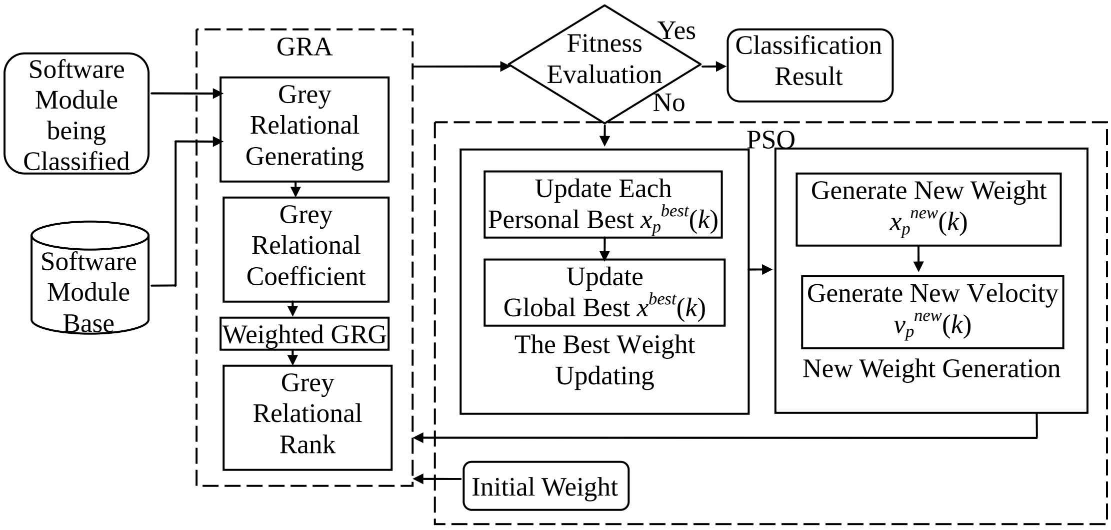

Fig. 1 A Framework of WGRA Approach

4. Experiments

A widely disseminated dataset of medical imaging system was utilized for evaluating the classification abilities [12]. For identifying the dependent variable of each module as *fp* (fault found) or *nfp* (not fault found) type, the modules without faults are considered as *nfp*, otherwise *fp*. That is, there are 51 *nfp* modules and 339 *fp* modules. For the independent variable, each module comprises with 11 software complexity metrics. For evaluation of software quality classification models, it may be too optimistic in the experiments if the model evaluation based on a particular training dataset. A cross-validation approach is a widely used method that may give a more objective assessment than a particular training dataset [1]. For evaluating software quality classification models, this study complies with the previous study that adopted the three-fold cross-validation approach.

In evaluating the proposed model of integration of GRA with PSO, two grey relational classifiers of the GRA with adopting the closest neighbor, e.g. GRA, and the GRA with adopting the k-nearest neighbors, e.g. NGRA, were applied to the above-mentioned datasets in order to compare with the proposed model of WGRA. The grey relational model with adopting the closest neighbor approach, e.g. GRA model, does not provide any adjustment mechanism for selecting a preferred balance model between *TIE* and *TIIE* rates for software quality classification because it always adopts the closest case as the classification result. For the grey relational model with adopting the k-nearest neighbors approach, e.g. NGRA model, it provides different numbers of nearest neighbors and adjust factor as an adjusting mechanism for providing a preferred balance model between *TIE* and *TIIE* rates for software quality classification. The other widely used software quality classification models of the C4.5 and CART [9] were also utilized to construct their respective software quality classification models for comparing the classification ability with the WGRA.

The classification results of GRA, NGRA, WGRA, C4.5 and CART approaches between training and test stages among three datasets of Fold-1, Fold-2 and Fold-3 are shown in Table 1. For the column “difference”, it shows the absolute difference between *TIE* and *TIIE* rates which is an indicator of a preferred balance model. A software quality classification model with a lower value of difference than the other models represents that it provides a good balance model between *TIE* and *TIIE* rates than the other models. Among three grey relational classifier models of GRA, NGRA and WGRA, applying PSO to GRA model of WGRA gives the lowest value of difference than the GRA and NGRA models at both the training and test stages on the Fold-1, Fold-2 and Fold-3 datasets. It also outperforms the CART and C4.5 approaches at both the training and test stages. For the classification accuracy, GRA model is slightly superior to the NGRA and WGRA models but the WGRA model somewhat outperforms the NGRA model. Generally, applying PSO to GRA can provide a preferred balance model between *TIE* and *TIIE* rates than the other models of GRA, NGRA, C4.5 and CART approaches based on difference evaluation criterion.

5. Discussion and Conclusion

The present study proposes the integration of PSO with grey relational classifier approach for software quality classification. The PSO approach is applied for investigating the suitable weights of grey

Dataset	Stage	Evaluation Criteria	Approach				
			GRA	NGRA	WGRA	C4.5	CART
Fold-1	Training	*TIE*	0.735	0.353	0.294	0.059	0.294
		TIIE	0.097	0.327	0.292	0.027	0.031
		Difference	0.638	0.026	0.002	0.032	0.263
		Accuracy	0.819	0.669	0.708	0.969	0.935
	Test	*TIE*	0.706	0.353	0.294	0.824	0.824
		TIIE	0.115	0.221	0.230	0.142	0.062
		Difference	0.591	0.132	0.064	0.682	0.762
		Accuracy	0.808	0.762	0.762	0.769	0.838
Fold-2	Training	*TIE*	0.706	0.324	0.294	0.059	0.588
		TIIE	0.093	0.310	0.292	0.044	0.027
		Difference	0.613	0.014	0.002	0.015	0.561
		Accuracy	0.827	0.688	0.708	0.954	0.900
	Test	*TIE*	0.647	0.412	0.353	0.471	0.882
		TIIE	0.106	0.354	0.336	0.080	0.071
		Difference	0.541	0.058	0.017	0.391	0.811
		Accuracy	0.823	0.638	0.662	0.869	0.823
Fold-3	Training	*TIE*	0.735	0.294	0.265	0.029	0.324
		TIIE	0.097	0.261	0.261	0.013	0.044
		Difference	0.638	0.033	0.004	0.016	0.280
		Accuracy	0.819	0.735	0.738	0.985	0.919
	Test	*TIE*	0.529	0.353	0.294	0.824	0.588
		TIIE	0.106	0.319	0.292	0.168	0.159
		Difference	0.423	0.034	0.002	0.656	0.429
		Accuracy	0.838	0.677	0.708	0.746	0.785

Table 1. Classification Results Among different Approaches

relational classifiers in order to supply a preferred model with balance of misclassification rate for software project managers. The experimental results are encouraging when PSO was applied on the grey relational classifiers to provide a balanced misclassification rate on the medical information system dataset. The improved GRA estimations, which integrated the PSO with grey relational classifiers, outperforms the grey relational classifiers without weighted approaches based on the evaluation criteria of gap between type I error rate and type II error rates. The above-mentioned results demonstrated that applying weights on software metrics to the grey relational classifiers is a feasible approach to yield a balanced misclassification rate for software project managers using the PSO approach. In addition, the proposed improved GRA is also comparable to the widely used estimation models of C4.5 and CART approaches.

We are encouraged by the results of the present study that the improved GRA provides a preferred balance on misclassification rates. This approach reveals a valuable guide for effect improvement the testing efforts by early identification of the suitable number of software modules which have high risk of faults as project managers required. The proposed approach has never been explored in the software engineering fields that offer a very promising technique in software quality classification. In the future, we believe that this approach can be extensively applied in many different fields for supporting decision makers. Furthermore, the investigation of classification abilities on the large and rigorous datasets, or another design of weights, e.g., linear or non-linear models, can be explored in the future works.

Acknowledgements

The author would like to thank the National Science Council (NSC) of Taiwan under the contract 96-2416-H-231-009 to support this research.

Reference

[1] N.H. Chiu, and S.J. Huang, The adjusted analogy-based software effort estimation based on similarity distances. *Journal of Systems and Software*, vol. 80, pp.628-640, 2007.

[2] J. Deng, Control problem of grey systems. *Systems and Control Letters*, vol. 1, no. 5, 1982.

[3] M.N.M. Garcia, I.R. Roman, F.J.G. Penalvo, and M.T. Bonilla, An association rule mining method for estimating the impact of project management policies on software quality development time and effort. *Expert Systems with Applications*, vol. 34, pp.522-529, 2008.

[4] S.J. Huang, N.H. Chiu, and L.W. Chen, Integration of the grey relational analysis with genetic algorithm for software effort estimation. *European Journal of Operational Research*, vol. 188, pp.898-909, 2008.

[5] J. Kennedy, and R. Eberhart, Particle swarm optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks, *Perth*, pp.1942-1948, 1995.

[6] M.J. Khan, S. Shamail, M.M. Awais, and T. Hussain, Comparative study of various artificial intelligence techniques to predict software quality. *IEEE Multitopic Conference*, pp.173-177, 2006.

[7] T.M. Khoshgoftaar, E.B. Allen, W.D. Jones, and J.P. Hudepohl, Classification tree models of software-quality over multiple releases. *IEEE Transactions on Reliability*, vol. 49, no. 1, pp.4-11, 2000.

[8] T.M. Khoshgoftaar, and N. Seliya, Analogy-based practical classification rules for software quality estimation. *Empirical Software Engineering*, vol. 8, pp.325-350, 2003.

[9] T.M. Khoshgoftaar, and N. Seliya, Comparative assessment of software quality classification techniques: an empirical case study. *Empirical Software Engineering*, vol. 9, pp.229-257, 2004.

[10] M. Lu, and K. Wevers, Application of grey relational analysis for evaluating road traffic safety measures: advanced driver assistance systems against infrastructure redesign. *Intelligent Transport Systems*, vol. 1, no. 1, pp.3-14, 2007.

[11] W.M. Lin, C.H. Huang, C.H. Lin, C.M. Chen, and L.W. Wang, Restoration strategy for secondary power network with grey relational analysis. Generation, Transmission and Distribution, IET, vol. 2, no. 2, pp.167-174, 2008.

[12] M.R. Lyu, Handbook of software reliability engineering. *IEEE Computer Society Press and McGraw-Hill Book Company*. 1996.

[13] I. Myrtveit, E. Stensrud, and M. Shepperd, Reliability and validity in comparative studies of software prediction models. *IEEE Transactions on Software Engineering*, vol. 31, no. 5, pp.380-391, 2005.

[14] M. Maitra, and A. Chatterjee, A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Systems with Applications*, vol. 34, pp.1341-1350, 2008.

[15] D. Rodriguez, R. Ruiz, J. Cuadrado-Gallego, J. Aguilar-Ruiz, and M. Garre, Attribute selection in software engineering datasets for detecting fault modules. *33rd EUROMICRO Conference on Software Engineering and Advanced Applications* (*SEAA 2007*), 2007.

[16] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, vol. 28, pp.459-471, 2007.

[17] J. Yisu, J. Knowles, L. Hongmei, L. Yizeng, and D.B. Kell, The landscape adaptive particle swarm optimizer. *Applied Soft Computing*, vol. 8, pp.295-304, 2008.