International Journal of Information Processing, 7(4), 87-101, 2013
ISSN : 0973-8215
IK International Publishing House Pvt. Ltd., New Delhi, India

Use Case Point Approach Based Software Effort Estimation using Various Support Vector Regression Kernel Methods

Shashank Mouli Satapathy[a], Santanu Kumar Rath[a]

[a]Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela 769008 India, Contact: shashankamouli@gmail.com, skrath@nitrkl.ac.in.

The job of software effort estimation is a critical one in the early stages of the software development life cycle when the details of requirements are usually not clearly identified. Various optimization techniques help in improving the accuracy of effort estimation. The Support Vector Regression (SVR) is one of several different soft-computing techniques that help in getting optimal estimated values. The idea of SVR is based upon the computation of a linear regression function in a high dimensional feature space where the input data are mapped via a nonlinear function. Further, the SVR kernel methods can be applied in transforming the input data and then based on these transformations, an optimal boundary between the possible outputs can be obtained. The main objective of the research work carried out in this paper is to estimate the software effort using use case point approach. The use case point approach relies on the use case diagram to estimate the size and effort of software projects. Then, an attempt has been made to optimize the results obtained from use case point analysis using various SVR kernel methods to achieve better prediction accuracy.

Keywords : Object Oriented Analysis and Design, Software Effort Estimation, Support Vector Regression, Use Case Point Approach.

1. INTRODUCTION

Proper software effort estimation is the foremost activity adopted in every software development life cycle. Several features offered by OO programming concept such as Encapsulation, Inheritance, Polymorphism, Abstraction, Cohesion and Coupling play an important role to manage the development process [1,2]. Currently used software development effort estimation models such as, COCOMO and Function Point Analysis (FPA), do not consistently provide accurate project cost and effort estimates [3]. These techniques have been proven unsatisfactory for estimating cost and effort because the lines of code (LOC) and function point (FP) are both used for procedural oriented paradigm [4]. Both of them have certain limitations. The LOC is dependent on the programming language and the FPA is based on human decisions. Hence effort estimation during early stage of software development life cycle plays a vital role for determining whether a project is feasible in terms of a cost-benefit analysis [5,6].

The Use Case Point (UCP) model relies on the use case diagram to estimate the effort of a given software product. UCP helps in providing more accurate effort estimation from design phase of software development life cycle. UCP is measured by counting the number of use cases and the number of actors, each multiplied by its complexity factors. Use cases and actors are classified into three categories. These include simple, average and complex. The determination of the complexity value (simple, average or complex) of use cases is determined by the number of transactions per use case. The UCP model has widely been used in the last decade [7], yet it has several limitations. One of the limitations is that the software effort equation is not well accepted by software estimators because it assumes that the relationship between software effort and size is linear. In this paper, various kernel methods based support vector regression is introduced to tackle the limitations of the UCP model and to enods. At the end of the study, a comparative analysis of the generated results has been presented in order to assess their accuracy.

While comparing the results with the results provided in the related work section, it can be concluded that the results obtained from proposed models outperform the results generated by models given in related work section. Similarly, while comparing the results obtained using various SVR kernel methods, it can be concluded that for UCP, RBF kernel based support vector regression technique outperformed other three kernel methods. The computations for above procedure have been implemented and membership functions generated using MATLAB. This can be further analyzed and proved if real data for Use Point Approach is available. This approach can also be extended by using some other soft computing techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

REFERENCES

1. M Carbone and G Santucci. Fast&&serious: a UML based Metric for Effort Estimation, In *Proceedings of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'02)*, pages 313–322, 2002.
2. Nasib S Gill and Sunil Sikka. New Complexity Model for Classes in Object Oriented System, *SIGSOFT Software Engineering Notes*, 35(5):1–7, October 2010.
3. Zhiwei Xu and Taghi M Khoshgoftaar. Identification of Fuzzy Models of Software Cost Estimation, *Fuzzy Sets and Systems*, 145(1):141–163, 2004.
4. J E Matson, B E Barrett and J M Mellichamp. Software Development Cost Estimation using Function Points, *IEEE Transactions on Software Engineering*, 20(4):275–287, 1994.
5. SG MacDonell and AR Gray. A Comparison of Modeling Techniques for Software Development Effort Prediction, In *Proceedings of Fourth International Conference on Neural Information Processing*, pages 869–872, Dunedin, New Zealand, 1998. Springer-Verlag.
6. Kevin Strike, Khaled El Emam and Nazim Madhavji. Software Cost Estimation with Incomplete Data, *IEEE Transactions on Software Engineering*, 27(10):890–908, 2001.
7. Mel Damodaran and A Washington. Estimation using Use Case Points, *Computer Science Program. Texas–Victoria: University of Houston. Sd*, 2002.
8. Ayman Issa, Mohammed Odeh and David Coward. Software Cost Estimation using Use-Case Models: A Critical Evaluation, In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 2, pages 2766–2771. IEEE, 2006.
9. Ali Bou Nassif, Danny Ho and Luiz Fernando Capretz. Regression Model for Software Effort Estimation based on the Use Case Point Method, In *Proceedings of International Conference on Computer and Software Modeling*, pages 117–121, 2011.
10. Ali Bou Nassif, Luiz Fernando Capretz and Danny Ho. A Regression Model with Mamdani Fuzzy Inference System for Early Software Effort Estimation based on Use Case Diagrams, In *Proceedings of Third International Conference on Intelligent Computing and Intelligent Systems*, pages 615–620, 2011.
11. Ali Bou Nassif, Luiz Fernando Capretz and Danny Ho. Estimating Software Effort based on Use Case Point Model using Sugeno Fuzzy Inference System, In *Proceedings of 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 393–398, 2011.
12. Ali Bou Nassif, Luiz Fernando Capretz and Danny Ho. Estimating Software Effort using an ANN Model based on Use Case Points, In *Proceedings of 11th IEEE International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, pages 42–47, 2012.
13. Ali Bou Nassif. Enhancing Use Case Points Estimation Method using Soft Computing Techniques, *Journal of Global Research in Computer Science*, 1(4), 2010.
14. Ali Bou Nassif, Luiz Fernando Capretz, Danny Ho and Mohammad Azzeh. A Treeboost Model for Software Effort Estimation based on Use Case Points, In *Proceedings of 11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, pages 314–319, 2012.
15. Adriano LI Oliveira. Estimation of Software Project Effort with Support Vector Regression, *Neurocomputing*, 69(13):1749–1753, 2006.
16. Petrônio L Braga, Adriano LI Oliveira and Silvio RL Meira. A Ga-based Feature Selection and Parameters Optimization for Support Vec-

tor Regression Applied to Software Effort Estimation, In *Proceedings of the 2008 ACM Symposium on Applied computing*, pages 1788–1792, 2008.
17. Ekrem Kocaguneli, Tim Menzies and Jacky W Keung. Kernel Methods for Software Effort Estimation, *Empirical Software Engineering*, 18(1):1–24, 2013.
18. Bilge Baskeles, Burak Turhan and Ayşe Bener. Software Effort Estimation using Machine Learning Methods, In *Proceedings of 22nd International symposium on Computer and Information Sciences*, pages 1–6, 2007.
19. Gustav Karner. Resource Estimation for Objectory Projects, *Objective Systems SF AB*, 17, 1993.
20. Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola and Vladimir Vapnik. Support Vector Regression Machines, *Advances in Neural Information Processing Systems*, pages 155–161, 1997.
21. Alex J Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression, *Statistics and Computing*, 14(3):199–222, 2004.
22. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011. Software Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
23. Tim Menzies, Zhihao Chen, Jairus Hihn, and Karen Lum. Selecting Best Practices for Effort Estimation, *IEEE Transactions on Software Engineering*, 32(11):883–895, 2006.

S M Satapathy did his MCA from Silicon Institute of Technology and M.Tech in Computer Science and Engineering from KIIT University, Bhubaneswar, India. Currently he is pursuing his Ph.D in Computer Science and Engineering at National Institute of Technology, Rourkela, India. His areas of interest are Software Cost Estimation, Program Slicing.

S K Rath is a Professor in the Department of Computer Science and Engineering, NIT Rourkela since 1988. His research interests are in Software Engineering, System Engineering, Bioinformatics and Management. He has published a large number of papers in international journals and conferences in these areas. He is a Senior Member of the IEEE, USA and ACM, USA and Petri Net Society, Germany.