Search-Based Predictive Modelling for Software Engineering: How Far Have We Gone?*

Federica Sarro

Department of Computer Science, University College London, f.sarro@ucl.ac.uk
http://www0.cs.ucl.ac.uk/staff/F.Sarro/

Abstract. In this keynote I introduce the use of Predictive Analytics for Software Engineering (SE) and then focus on the use of search-based heuristics to tackle long-standing SE prediction problems including (but not limited to) software development effort estimation and software defect prediction. I review recent research in Search-Based Predictive Modelling for SE in order to assess the maturity of the field and point out promising research directions. I conclude my keynote by discussing best practices for a rigorous and realistic empirical evaluation of search-based predictive models, a *condicio sine qua non* to facilitate the adoption of prediction models in software industry practices.

Keywords: Predictive Analytics · Predictive Modelling · Search-Based Software Engineering · Machine Learning · Software Analytics

1 Introduction

Nowadays software pervades almost every aspect of our life. This allows the production and collection of a large amount of information about people's decisions and behaviours. Predictive Analytics exploits such information through intelligent systems which are able to identify patterns and predict future outcomes and trends. Applied to Software Engineering, predictive analytics helps us better understand software processes, products and customers in order to maximise product quality, users satisfaction, and revenues [27].

One of the most important use of Predictive Analytics for Software Engineering is building prediction systems to estimate crucial software aspects and support engineers throughout the software production life-cycle (a.k.a Predictive Modelling for Software Engineering). Examples of software engineering prediction problems are: estimating the amount of effort likely required to develop or maintain software [25, 11], estimating the successes of mobile applications [29] and identifying software that will most likely contain defects [12], cause crashes [33] or fail tests [19].

Predictive Modelling for Software Engineering has been an important and active research field that can be dated back to 1971, when the first attempt to

* This paper provides an outline of the keynote talk given by Dr. Federica Sarro at SSBSE 2019, with pointers to the literature for details of the results covered.

estimate the number of software defects was made [18]. Since then, predictive systems of various nature have been proposed ranging from statistical models and analogy-based techniques to machine learning and search-based methods. In particular, over the past 10 years, search-based prediction systems have been specifically devised to tackle long-standing software engineering prediction problems such as software development effort, defect proneness, maintainability and change proneness [14, 21]. These systems are either stand-alone systems able to build optimal prediction models [6, 10, 25] or ones that are used in combination with other (usually machine learning-based) estimators [3–5, 20, 24]. A variety of meta-heuristics based on both local and global search techniques (e.g., Simulated Annealing, Tabu Search, Genetic Algorithm, Genetic Programming) has been used, with the latter being definitively the most studied [8, 21, 26] and with Multi-Objective Evolutionary Algorithm usually resulting in the most effective approach for different prediction tasks (see e.g. [2, 25]).

In this keynote I explain how to use search-based heuristics to tackle software engineering prediction problems. I also highlight their strengths and weaknesses with respect to more traditional statistical or machine learning-based estimators. Some of these are the possibility to use one or multiple desired measures as a fitness function to evolve optimal prediction models [2, 7, 25, 28] and the need of scalable solutions [9, 23]. I review the most promising results in this field and also envisage novel applications of search-based heuristics to predictive modelling for SE; this includes using them to analyse interesting trade-offs (e.g. models' predictive quality vs. interpretability) and to test machine learning-based predictors, both of which are challenges currently faced by the wider SE community. I conclude my keynote by discussing best practices for a rigorous [1, 16, 22, 30, 31] and realistic [13, 15, 17, 32] empirical assessment and evaluation of search-based predictive models, which is a *condicio sine qua non* to grow this field and to facilitate the adoption of prediction models in software industry practices.

References

1. Arcuri, A., Briand, L.C.: A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. STVR **24**(3), 219–250 (2014)
2. Canfora, G., De Lucia, A., Di Penta, M., Oliveto, R., Panichella, A., Panichella, S.: Multi-objective cross-project defect prediction. In: Procs. of the IEEE 6th International Conference on Software Testing, Verification and Validation. pp. 252–261. ICST'13 (2013). https://doi.org/10.1109/ICST.2013.38
3. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E.: How effective is tabu search to configure support vector regression for effort estimation? In: Proc. of the International Conference on Predictive Models in Software Engineering. pp. 4:1–4:10. PROMISE'10 (2010). https://doi.org/10.1145/1868328.1868335
4. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E.: Using tabu search to configure support vector regression for effort estimation. Empirical Software Engineering **18**(3), 506–546 (2013). https://doi.org/10.1007/s10664-011-9187-3

5. Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F.: A genetic algorithm to configure support vector machines for predicting fault-prone components. In: Product-Focused Software Process Improvement. pp. 247–261. Springer (2011). https://doi.org/10.1007/978-3-642-21843-9_20
6. Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F.: Using tabu search to estimate software development effort. In: Proc. of the International Workshop on Software Measurement. pp. 307–320. MENSURA'09, LNCS 5891, Springer (2009). https://doi.org/10.1007/978-3-642-05415-0_22
7. Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F.: Genetic programming for effort estimation: An analysis of the impact of different fitness functions. In: Proc. of the 2nd International Symposium on Search Based Software Engineering. pp. 89–98. SSBSE'10 (2010). https://doi.org/10.1109/SSBSE.2010.20
8. Ferrucci, F., Harman, M., Sarro, F.: Search-based software project management. In: Software Project Management in a Changing World, pp. 373–399. Springer (2014)
9. Ferrucci, F., Salza, P., Sarro, F.: Using hadoop mapreduce for parallel genetic algorithms: A comparison of the global, grid and island models. Evolutionary Computation pp. 1–33 (2017). https://doi.org/10.1162/evco_a_00213
10. Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F., Mendes, E.: Investigating tabu search for web effort estimation. In: Procs. of EUROMICRO Conference on Software Engineering and Advanced Applications. pp. 350–357. SEAA'10 (2010)
11. Ferrucci, F., Mendes, E., Sarro, F.: Web effort estimation: The value of cross-company data set compared to single-company data set. In: Proceedings of the 8th International Conference on Predictive Models in Software Engineering. pp. 29–38. ACM (2012)
12. Hall, T., Beecham, S., Bowes, D., Gray, D., Counsell, S.: A systematic literature review on fault prediction performance in software engineering. IEEE Trans. Softw. Eng. **38**(6), 1276–1304 (2012). https://doi.org/10.1109/TSE.2011.103
13. Harman, M., Islam, S., Jia, Y., Minku, L.L., Sarro, F., Srivisut, K.: Less is more: Temporal fault predictive performance over multiple hadoop releases. In: Procs. of the International Symposium on Search-Based Software Engineering (SSBSE'14). pp. 240–246. Springer (2014). https://doi.org/10.1007/978-3-319-09940-8_19
14. Harman, M.: The relationship between search based software engineering and predictive modeling. In: Procs. of the 6th International Conference on Predictive Models in Software Engineering. pp. 1:1–1:13. PROMISE'10 (2010). https://doi.org/10.1145/1868328.1868330
15. Jimenez, M., Rwemalika, R., Papadakis, M., Sarro, F., Le Traon, Y., Harman, M.: The importance of accounting for real-world labelling when predicting software vulnerabilities. In: Procs. of the 27th ACM SIGSOFT International Symposium on the Foundations of Software Engineering. ESEC/FSE'19 (2019)
16. Langdon, W.B., Dolado, J.J., Sarro, F., Harman, M.: Exact mean absolute error of baseline predictor, MARP0. Inf. Softw. Technol. **73**, 16–18 (2016). https://doi.org/10.1016/j.infsof.2016.01.003
17. Lanza, M., Mocci, A., Ponzanelli, L.: The tragedy of defect prediction, prince of empirical software engineering research. IEEE Software **33**(6), 102–105 (2016). https://doi.org/10.1109/MS.2016.156
18. Menzies, T., Zimmermann, T.: Software analytics: So what? IEEE Software **30**(4), 31–37 (2013). https://doi.org/10.1109/MS.2013.86
19. Najafi, A., Rigby, P., Shang, W.: Bisecting commits and modeling commit risk during testing. In: Procs. of the 27th ACM SIGSOFT International Symposium on the Foundations of Software Engineering. ESEC/FSE'19 (2019)

20. P. Braga, L., A. Oliveira, L.I., S. Meira, R.L.: A ga-based feature selection and parameters optimization for support vector regression applied to software effort estimation. In: Proc. of the ACM Symposium on Applied Computing. pp. 1788–1792. SAC'08 (2008)
21. Ruchika, M., Megha, K., Rajeev, R.R.: On the application of search-based techniques for software engineering predictive modeling: A systematic review and future directions. Swarm and Evolutionary Computation **32**, 85 – 109 (2017)
22. Russo, B.: A proposed method to evaluate and compare fault predictions across studies. In: Procs. of the 10th International Conference on Predictive Models in Software Engineering. pp. 2–11. PROMISE'14, ACM (2014). https://doi.org/10.1145/2639490.2639504
23. Salza, P., Ferrucci, F., Sarro, F.: Elephant56: Design and implementation of a parallel genetic algorithms framework on hadoop mapreduce. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. pp. 1315–1322. GECCO'16 (2016). https://doi.org/10.1145/2908961.2931722
24. Sarro, F., Di Martino, S., Ferrucci, F., Gravino, C.: A further analysis on the use of genetic algorithm to configure support vector machines for inter-release fault prediction. In: Procs. of the 27th Annual ACM Symposium on Applied Computing. pp. 1215–1220. SAC'12 (2012). https://doi.org/10.1145/2245276.2231967
25. Sarro, F., Petrozziello, A., Harman, M.: Multi-objective software effort estimation. In: Procs.of the 38th International Conference on Software Engineering. pp. 619–630. ICSE'16 (2016). https://doi.org/10.1145/2884781.2884830
26. Sarro, F.: Search-based approaches for software development effort estimation. In: Procs. of the 12th International Conference on Product Focused Software Development and Process Improvement. pp. 38–43. PROFES'11 (2011). https://doi.org/10.1145/2181101.2181111
27. Sarro, F.: Predictive analytics for software testing: Keynote paper. In: Proceedings of the 11th International Workshop on Search-Based Software Testing. pp. 1–1. SBST'18 (2018). https://doi.org/10.1145/3194718.3194730
28. Sarro, F., Ferrucci, F., Gravino, C.: Single and multi objective genetic programming for software development effort estimation. In: Procs. of the 27th Annual ACM Symposium on Applied Computing. pp. 1221–1226. SAC'12 (2012). https://doi.org/10.1145/2245276.2231968
29. Sarro, F., Harman, M., Jia, Y., Zhang, Y.: Customer rating reactions can be predicted purely using app features. In: Procs. of 26th IEEE International Requirements Engineering Conference. pp. 76–87. RE'18 (2018). https://doi.org/10.1109/RE.2018.00018
30. Sarro, F., Petrozziello, A.: Linear programming as a baseline for software effort estimation. ACM Trans. Softw. Eng. Methodol. **27**(3), 12:1–12:28 (2018). https://doi.org/10.1145/3234940
31. Shepperd, M.J., MacDonell, S.G.: Evaluating prediction systems in software project estimation. Information and Sofware Technology **54**(8), 820–827 (2012). https://doi.org/10.1016/j.infsof.2011.12.008
32. Sigweni, B., Shepperd, M., Turchi, T.: Realistic assessment of software effort estimation models. In: Procs. of the 20th International Conference on Evaluation and Assessment in Software Engineering. pp. 41:1–41:6. EASE'16, ACM (2016). https://doi.org/10.1145/2915970.2916005
33. Xia, X., Shihab, E., Kamei, Y., Lo, D., Wang, X.: Predicting crashing releases of mobile applications. In: Procs. of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 29:1–29:10. ESEM'16 (2016). https://doi.org/10.1145/2961111.2962606