Learning to Generate Fault-revealing Test Cases in Metamorphic Testing

Helge Spieker[1] Arnaud Gotlieb[1]

Abstract: Metamorphic Testing is a software testing paradigm which aims at using necessary properties of a system under test, called metamorphic relations (MR), to either check its expected outputs, or to generate new test cases [Se16]. Metamorphic Testing has been successful to test programs for which a full oracle is unavailable or to test programs with uncertainties on expected outputs such as learning systems. In this paper, we formulate the effective selection of MRs as a reinforcement learning problem, based on *contextual bandits*. Our method *Adaptive Metamorphic Testing* sequentially selects a MR that is expected to provide the highest payoff, i.e., that is most likely to reveal faults. Which MRs are likely to reveal faults is learned from successive exploration trials. The bandit explores the available MRs and evaluates the fault landscape of the system under test, thereby providing valuable information to the tester. We present experimental results over two applications in machine learning, namely image classification and object detection, where Adaptive Metamorphic Testing efficiently identifies weaknesses of the tested systems. The original paper "Adaptive Metamorphic Testing with Contextual Bandits" first appeared in the Journal of Systems and Software (2020) [SG20].

Keywords: Software Testing; Metamorphic Testing; Machine Learning; Contextual Bandits

Metamorphic testing (MT) is a testing paradigm, in which a source test case is transformed into a new follow-up test case for which the exact expected outcome is unknown, but a relation between the source and follow-up test case is available [CCY98, Ch18]. MT aims at using necessary properties of a software under test to either check its expected outputs or to generate new test cases. By execution of the follow-up test case it can be confirmed whether the system-under-test behaves according to the so-called metamorphic relation. If the relation is violated, a failure in the system has been identified. Metamorphic testing thereby addresses the oracle problem in software testing, where it is impossible or difficult to know the exact system output for a test case.

Typical examples for successful applications include machine learning models used for classification tasks, for which only stochastic behaviors can be specified [Ba15]. Indeed, these models are often initially trained with existing datasets and then exploited to classify new data samples. However, the expected class of any new data sample is unknown and thus, these samples cannot be used for testing the trained models. Fortunately, transformations over the data samples which do not change their (unknown) class, are usually available. By applying these transformations, i.e. the metamorphic relations (MRs) in MT, it becomes possible to effectively test machine learning models and their training [Mu08, DHG17].

[1] Simula Research Laboratory, Norway {helge,arnaud}@simula.no

doi:10.18420/SE2021_37

This papers addresses the problem of MR selection, i.e. determining which from a set of known MRs are best suited to discover faults in the system under test. We formulate the effective selection of MRs as a reinforcement learning problem, based on *contextual bandits* [LZ07]. Our method *Adaptive Metamorphic Testing* (AMT) defines a *test transformation bandit* which sequentially selects a MR that is expected to provide highest payoff, i.e., is most likely to reveal faults. Which MRs are likely to reveal faults is learned from successive exploration trials. The bandit explores the different MRs and evaluates the fault landscape of the system under test, thereby providing valuable information to the tester.

Learning the selection of MRs can be useful when testing under resource-constraints, for example in cases where the system under test changes are frequently integrated and tested, but also for infrequent testing when the number of MRs is large or their checking is costly. We have applied our method to test deep learning systems for computer vision. Adaptive metamorphic testing showed to find the same failure rate and distribution than exhaustive testing while requiring less test executions, which can be costly or at least time-consuming. At the same time, it's error discovery is stronger than pure random testing, because it can adapt to the strengths of the available metamorphic relations for certain test cases. Our implementation and datasets are available at: `https://github.com/HelgeS/tetraband`

Bibliography

[Ba15] Barr, Earl T.; Harman, Mark; McMinn, Phil; Shahbaz, Muzammil; Yoo, Shin: The Oracle Problem in Software Testing: A Survey. IEEE Transactions on Software Engineering, 41(5):507–525, 2015.

[CCY98] Chen, T.Y.; Cheung, S.C.; Yiu, S.M.: Metamorphic Testing: A New Approach for Generating Next Test Cases. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, 1998.

[Ch18] Chen, Tsong Yueh; Kuo, Fei-Ching; Liu, Huai; Poon, Pak-Lok; Towey, Dave; Tse, T. H.; Zhou, Zhi Quan: Metamorphic Testing: A Review of Challenges and Opportunities. ACM Computing Surveys, 51(1), 2018.

[DHG17] Ding, Junhua; Hu, Xin-Hua; Gudivada, Venkat: A Machine Learning Based Framework for Verification and Validation of Massive Scale Image Data. IEEE Transactions on Big Data, 26(3):1–1, 2017.

[LZ07] Langford, John; Zhang, Tong: The Epoch-Greedy Algorithm for Multi-Armed Bandits with Side Information. In: Advances in Neural Information Processing Systems 20 (NIPS 2007). pp. 817–824, 2007.

[Mu08] Murphy, Christian; Kaiser, Gail; Hu, Lifeng; Wu, Leon: Properties of Machine Learning Applications for Use in Metamorphic Testing. Proc. of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 867–872, 2008.

[Se16] Segura, Sergio; Fraser, Gordon; Sanchez, Ana B.; Ruiz-Cortes, Antonio: A Survey on Metamorphic Testing. IEEE Transactions on Software Engineering, 42(9):805–824, 2016.

[SG20] Spieker, Helge; Gotlieb, Arnaud: Adaptive Metamorphic Testing with Contextual Bandits. Journal of Systems and Software, 165:110574, July 2020.