Accurate Modeling of Performance Histories for Evolving Software Systems

Stefan Mühlbauer[1], Sven Apel[2], Norbert Siegmund[3]

Keywords: Software Performance; Software Evolution; Test Prioritization

1 Introduction Throughout a software system's development history, its non-functional properties, such as performance, evolve alongside. Individual modifications of the code base (*revisions*) or batches thereof can entail changes in performance. Unless identified and addressed, detrimental performance changes can add up to performance degrading over time. The retrospective analysis of existing histories can unveil causative revisions and, subsequently, help prioritize revisions for future performance regression testing. As performance measurements come at a considerable cost, it is intractable to assess all revisions. Instead, the challenge is to find a trade-off between measurement effort and accuracy of estimating performance.

We devise a novel probabilistic *active learning* algorithm to *accurately* approximate the performance history of a software system based on measurements of a specific workload with *accurately with few measurements* [MAS19]. Our approach is not only able to provide performance estimations for all revisions, but also reports an uncertainty measure alongside. We use this uncertainty measure to decide for each revision whether our estimation is sufficiently accurate or whether we need to refine the approximation by including more measurements. To increase reliability where necessary, the algorithm selects and prioritizes new revisions for performance measurement based on the reported uncertainty and relearns the underlying Gaussian Process model.

Our evaluation is based on six real-world software systems from a variety of domains (file compression, scientific computing, image processing). In a preliminary analysis, we confirmed the prevalence of performance change points throughout their development histories. Our experiments show that we can approximate performance histories with high accuracy. We can use these to identify performance change points with few measurements.

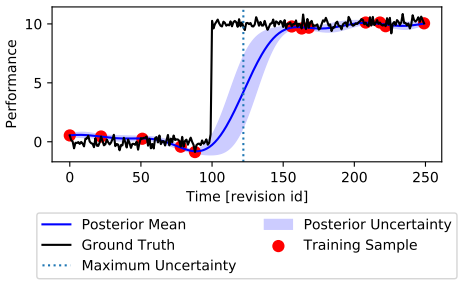
[1] Leipzig University, Institute of Computer Science, Augustusplatz 10, 04109 Leipzig, Germany, muehlbauer@informatik.uni-leipzig.de
[2] Universität des Saarlandes, Saarland Informatics Campus, Campus E1.1, 66123 Saarbrücken, Germany, apel@cs.uni-saarland.de
[3] Leipzig University, Institute of Computer Science, Augustusplatz 10, 04109 Leipzig, Germany, norbert.siegmund@uni-leipzig.de

2 Approach. Based on a small initial sample of measured revisions, we approximate a performance history following two repeated steps: (a) performance history estimation with Gaussian Processes and (b) active revision sampling. Next, we give an overview of both two steps:

Approximating Performance Histories. We use Gaussian Processes (GPs) for time series data as a framework to model the performance history of a software system and obtain respective estimations. In a nutshell, a GP can be conceived as a distribution over functions (here: performance as a function of revisions). Evaluating the GP for a revision will yield a Gaussian $\mathcal{N}(\mu, \sigma)$ with a mean performance estimate μ and a variance measure σ. The variance σ is lower around revisions for which we have actual performance measurements at hand and can be interpreted as a measure of prediction accuracy. The shape of an approximated performance history is determined by the GP's covariance function – a hyper parameter often called *kernel*. The kernel encodes further properties of the modeled performance histories, such as whether to expect a continuous estimation.

Active Revision Sampling. At large, we evaluate the GP for all revisions to obtain an approximation as in Fig. 1 with regions of low and high uncertainty indicating the need for further measurements. The key idea of our approach is the following: We let the uncertainty measures *guide* the selection of new revisions to measure performance for. That is, we interpret the prediction uncertainty as a measure of how much we expect this measurement to improve overall prediction accuracy. We repeat these two model refinement steps until the minimum uncertainty across all revisions falls below a user-specified threshold.

Fig. 1: GP estimation of a performance history with one change point (at revision 120) based on ten measurements.



We perform a series of experiments with the six real-world subject systems XZ, LRZIP, PILLOW, ULTRAJSON, NUMPY and SCIPY. Across different covariance functions evaluated, we obtained the most accurate approximations of performance histories in setups with the Brownian kernel. From such approximations, we are able to identify and pinpoint change points to individual revisions.

Bibliography

[MAS19] Mühlbauer, Stefan; Apel, Sven; Siegmund, Norbert: Accurate Modeling of Performance Histories for Evolving Software Systems. In: Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, pp. 640–652, 2019.