Abstract

The core idea of this thesis is to reduce workload of manual inspection when the performance analysis of an updated software is required. CPU utilization, which is one of the essential factors for evaluating the performance, is analyzed. The purpose of this work is to apply machine learning techniques that are suitable for detecting the state of the CPU utilization, and also any changes in the test environment that effects the CPU utilization. The detection relies on a Markov switching model to identify structural changes, which are assumed to follow an unobserved Markov chain, in the time series data. A historical behavior of the data can be described by a first-order autoregression. Then, the Markov switching model becomes a Markov switching autoregressive model. Another approach based on a non-parametric analysis, a distribution-free method that requires fewer assumptions, called an E-divisive method is purposed. This method used hierarchical clustering algorithm to detect multiple change point locations in the time series data. As the data used in this analysis does not contain any ground truths, the evaluation of the methods is analyzed by generating simulated datasets with known states. Besides, these simulated datasets are used for studying and comparing between the Markov switching model and the E-divisive method. Results show that the former method is preferable because of its better performance and more efficiency in detecting changes. The E-divisive method can detect fewer changes and has a higher rate of missed detections. The results from applying the Markov switching autoregressive model to the real data are presented with interpretations and discussions.

Keywords: Markov switching model, Non-parametric analysis, CPU utilization