Apply machine learning to Performance trend analysis

Araya Eamrurksiri

May 19, 2017

- Many test cases are executed for testing software packages
- Evaluate how each software package performs for an updated software package
- Tool or algorithm that can reduce workload of manual inspection

Objectives

- Detect the state of the CPU utilization (degrading, improving or steady state)
- Detect whether there is any change in the test environment that effects the CPU utilization

- Software release
- Software package - treated as a time point in the time series

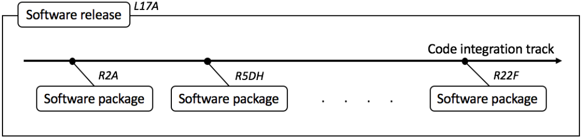

Figure: Several software packages that are launched in the timeline

- Test cases in QA capacity area on signaling capacity - treated as an observation in the dataset

Data is collected on January 20, 2017

Three datasets: Software release L16A, L16B, and L17B

- Sorted by software package version
- Filtered out test cases which are not executed properly
- Selected test case which has *lowest* value of the CPU utilization to represent a performance of a specific software package

In total, each dataset contains 64, 241, and 144 test cases, respectively

EventsPerSec: Event intensity

- Contains several *local events*
- Stores multiple values separated by a tab character
- Some local events are used as predictor variables
- Implement a function to split each element to columns

Response variable

- TotCpu%: CPU utilization

Predictor vairalbes

- EventsPerSec
    - RrcConnectionSetupComplete
    - Paging
    - X2HandoverRequest
- Test environments
    - DuProdName: Product hardware name
    - Fdd/Tdd: Different standard of LTE 4G Technology
    - NumCells: Number of cells in the base station

Markov switching model [Hamilton, 1989]
Assuming that S_t denote an unobservable state variable

$$y_t = X_t\beta_{S_t} + \phi_{1,S_t}y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2_{S_t})$$

y_t is the observed value of time series at time t
X_t are the predictor variables of time series at time t
β_{S_t} are the coefficients in state S_t, where $S_t = 1, 2, ..., k$
ϕ_{1,S_t} is an autoregression coefficient of the observed value at time $t-1$ in state S_t
The observation are drawn from the first order autoregressive model, AR(1).

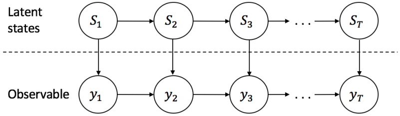

Figure: Model structure

Markov switching model

df

When applying the Markov switching model, we need to decide on

- Number of states, k
- Number of switching coefficients in the model

Based on the applied literature, the information criteria called the Bayesian Information Criterion is used to select these numbers

$$\mathrm{BIC} = -2\ln(L(\hat{\theta})) + m \cdot \ln(T)$$

E-divisive method [James, 2016]

- Non-parametric approach: more flexible as no assumption about the distribution is made
- Detects multiple change point locations based on a divisive hierarchical estimation algorithm
- Algorithm:
  recursively partition a time series, and perform a permutation test to find the statistical significance of an estimated change point.

- State of the CPU utilization is unknown

- State of the CPU utilization is unknown
- Simulated two datasets - Dataset 1 and Dataset 2 - with different switching between states

$$y_t = \begin{cases} 10 + 0.6X_{1,t} - 0.9X_{2,t} + 0.5y_{t-1} + \varepsilon_t^{(1)} & \text{Normal} \\ 2 + 0.8X_{1,t} + 0.2y_{t-1} + \varepsilon_t^{(2)} & \text{Bad} \\ -12 + 0.7X_{1,t} + 0.2X_{2.t} - 0.2y_{t-1} + \varepsilon_t^{(3)} & \text{Good} \end{cases}$$

y_t is assumed to be a value of a CPU usage of the time series at time t
$x_{1,t} \sim U[50, 200]$ of the time series at time t
$x_{2,t} \sim U[0, 50]$ of the time series at time t
$\varepsilon_t^{(1)} \sim N(0, 1), \quad \varepsilon_t^{(2)} \sim N(2, 0.5), \quad \text{and} \quad \varepsilon_t^{(3)} \sim N(1, 1)$

Decide: Number of states
Hypothesis: Markov switching model with *two* or *three* states

---

- BIC is one criteria to select the appropriate model but model output and plot should also be taken into account
- **Three-state** model are chosen for further analysis
- Remark:
  Higher number of states $k \geq 4$ are more likely to give worse results and were not considered

Analysis II: Number of switching coefficients

Decide: Number of switching coefficients in the model
Hypothesis: test environments is possible to have non-switching effects

---

Software release L16A
Model with Fdd/Tdd and Numcells are non-switching coefficients

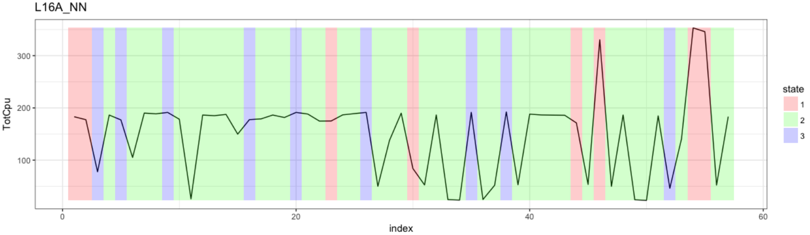

Figure: The CPU utilization showing the periods where the observation is in the specific state.

Decide: Number of switching coefficients in the model
Hypothesis: test environments is possible to have non-switching effects

---

Software release L16B

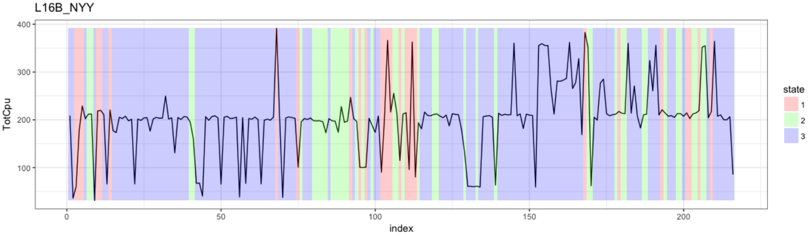

Figure: The CPU utilization showing the periods where the observation is in the specific state.

Decide: Number of switching coefficients in the model
Hypothesis: test environments is possible to have non-switching effects

---

Software release L17A

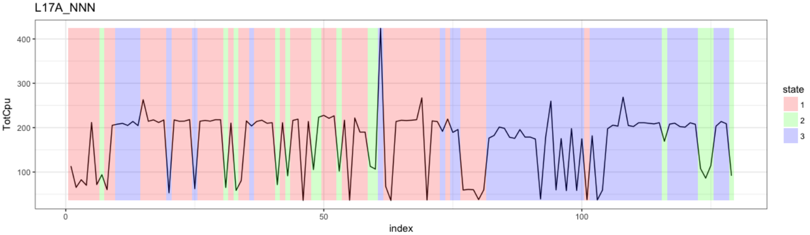

Figure: The CPU utilization showing the periods where the observation is in the specific state.

Simulated Dataset 1

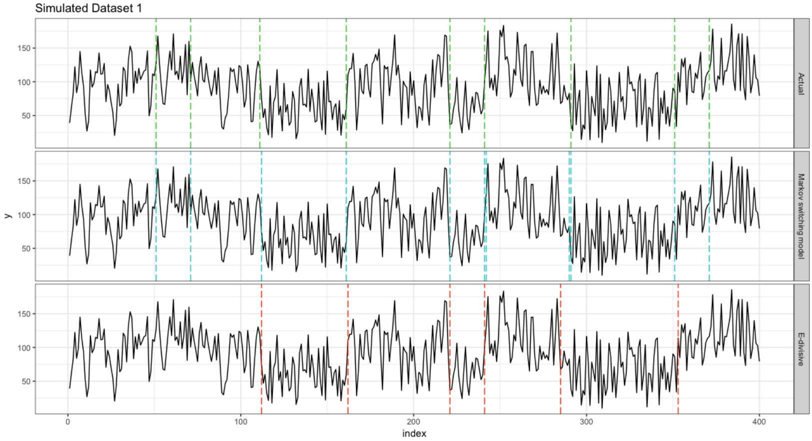

Figure: The simulated Dataset 1 showing the estimated change point locations

Simulated Dataset 2

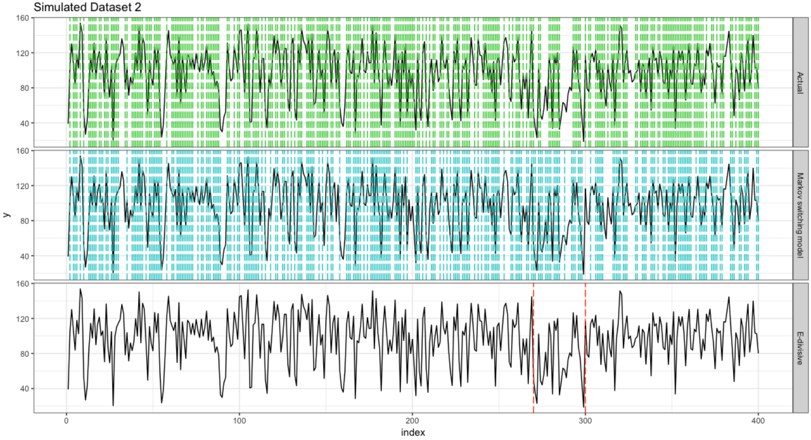

Figure: The simulated Dataset 2 showing the estimated change point locations

Real data: Software release L16A

Real data: Software release L16B

Real data: Software release L17A

Concludeeeee

- Larger dataset
- Effects of other variables
- Consider on the other performance metrics (e.g.,memory usage and latency)
- Use semi-supervised learning algorithm if some test cases are labeled with state

References

James D Hamilton (1989)
A new approach to the economic analysis of nonstationary time series and the business cycle
Econometrica: Journal of the Econometric Society, pages 357-384.

Josep A. Sanchez-Espigares and Alberto Lopez-Moreno (2014)
MSwM: Fitting Markov Switching Models
CRAN R.

Nicholas A. James and David S. Matteson (2016)
ecp: Nonparametric Multiple Change Point Analysis of Multivariate Data
CRAN R.