Masarykova univerzita
Fakulta informatiky

Alert prediction in metric data based on machine learning methods

Master thesis

Pavol Loffay

Brno, spring 2016

Declaration

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Pavol Loffay

Advisor: RNDr. Adam Rambousek

Acknowledgement

I would like to thank my supervisor Jiri Kremser, family and the people from Hawkular team. Next, my thanks goes to Ing. Daniel Němec, Ph.D. and Ing. Daniel Schwarz, Ph.D. for consulting theory behind time series analysis.

Last but not least, I would like to thank company Red Hat, that provided me the great opportunity to work on this project.

Abstract

The aim of the master thesis was to develop a module for open source monitoring and management platform Hawkular. This module is responsible for predicting alerts based on time series.

Keywords

Time Series, Hawkular, Alert Prediction, ARIMA, LMS

Contents

1 Introduction

An Alert prediction is very important because it can predict critical states of a system in advance. It gives system administrators powerful feature to reduce for instance downtime of an application or ability to load balance workload in advance. TODO

1.1 Context

The implementation part of the master thesis is developed as part of the project Hawkular[1]. Therefore the application architecture and used technologies had to fit into the overall project environment.

Hawkular is an open source monitoring and management platform mainly developed by company Red Hat. It is successor of very successful RHQ project[2]. This application is designed to monitor Java application servers like Wildfly, Apache Tomcat and other middleware systems. For instance it can monitor heap usage, web sessions, used data sources etc.

TODO describe more Hawkular - modules(mention scalability), metrics structure

1.2 Goals

The main goal of the application is to provide users with reliable predictions of alerts for the set of collected metrics. Displaying of forecast is also very important and should be available in the user interface. This feature can help system administrators react on events like running out of memory well in advance.

TODO performance(loads of metrics),

1. Available at <`http://www.hawkular.org`>
2. Available at <`https://rhq-project.github.io/rhq/`>

2 Time Series Analysis

In this chapter are discussed various approaches for forecasting time series and also relevant theory which is helpful for understanding them. Models are ordered from simpler to more complex ones.

In the early part of the thesis development system R was used for quickly showing the results of forecasts and also for better understanding metric data by plotting it.

Firstly it is important to define time series; it is sequence of observations $s_t \in \mathbb{R}$ usually ordered in time. In this thesis are used only equidistant discrete univariate time series.

Forecasting is a process of making prediction of the future based on the past. In other words forecasting is possible because future depends on the past or analogously because there is a relationship between the future and the past. However this relation is not deterministic and can be hardly written in an analytical form.

There are two forecasting types: qualitative and quantitative. Qualitative methods are mainly based on the opinion of the subject and are used when past data are not available, hence not suitable for this project. When there are past data available quantitative forecasting methods are more suitable.

2.1 Simple quantitative methods

There are three simple quantitative forecasting methods:

- Average method – forecasts are equal to the value of the mean of historical data.

$$\hat{y}_{T+h|T} = \overline{y} = (y_1 + \cdots + y_T)/T$$

- Naïve method – forecasts are equal to the last observed value.

$$\hat{y}_{T+h|T} = y_T$$

- Drift method – variation of naïve method which allow the forecasts to increase or decrease over time.

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T} y_t - y_{t-1} = y_T + h(\frac{y_T - y_1}{T-1})$$

There is also an seasonal variant of naïve method. This method is suitable only for highly seasonal data. Forecast is simply equal to last observed value from the previous season.

2.2 Time series decomposition

In the time series can be seen various patterns(TODO ref otext). It is crucial to categorize some of them. Basic observed patterns are trend, seasonality, cycle and irregular component also called white noise.

- **Trend** T_t – exists if there is long term increase or decrease over time. Can be linear or nonlinear (e.g. exponential growth)
- **Seasonal** S_t – exists when a series is influenced by seasonal factors. Seasonality is always of fixed and known period.
- **Cyclic** C_t – exists it there are long term wave – like patterns. Unlike trend waves are not of a fixed period.
- **Irregular** N_t – unpredictable random value referred as white noise.

Decomposition can be in many forms for instance two of them are additive and multiplicative model.

$$y_t = T_t + S_t + C_t + N_t \tag{2.1}$$
$$y_t = T_t \times S_t \times C_t \times N_t \tag{2.2}$$

2.3 Averaging and smoothing models

2.3.1 Moving Average Smoothing

This model can eliminate some randomness in the data

2.3.2 Exponential Smoothing

The concept behind simple exponential smoothing is to attach larger weights to the most recent observations than to observations from distant past. Forecasts are calculated using weighted averages where the weights decrease exponentially as observations come from further in the past. In other words smaller weights are associated to older observations. Equation for simple exponential smoothing is listed in 2.3.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \dots \tag{2.3}$$

Smoothing parameter is $0 \leq \alpha \leq 1$. Note, if $\alpha = 1$ then $\hat{y}_{T+1|T} = y_T$ so forecasts are equal to the naïve method. If the parameter α is smaller more weight is given to observations from distance in past.

Simple exponential smoothing has flat forecast function, that means all forecast all the same. Smoothing can be generally used as technique to separate signal and noise. This method is useful if a series doesn't contain any trend.

Holt's Liner Trend Method

Simple exponential smoothing can be extended to allow forecasting of data with a trend. This was done by Charles C. Holt in 1957. This method is slightly more complicated than original one without trend. In order to add trend component another equation has to be added.

$$\begin{aligned} \hat{y}_{t+h|t} &= l_t + hb_t \\ l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \end{aligned} \tag{2.4}$$

Where a parameter b_t denotes a slope of the series and a parameter l_t level. There is also a new parameter smoothing parameter of the slope $-\beta$. It's rage is equal to α, so $\alpha, \beta \in [0,1]$.

2.4 Linear Regression

2.5 Box – Jenkins Methodology

Methods from Box – Jenkins methodology are the most widely used in the time series modelling. It analyzes autocorrelations (ACF) and partial autocorrelations (PACF) between lagged observations. This two functions are used to analyze time series and estimate model parameters.

2.5.1 Autoregressive models (AR)

Forecast in an autoregressive model is done by linear combination of past values of the observed variable. Basically it is a linear regression of the current value of the time series against prior values of the series.

An autoregressive model of order p can by written as

$$y_t = c + e_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} \tag{2.5}$$

This model is also known as $AR(p)$ model, where p is the order of the AR model.

2.5.2 Moving average models (MA)

Similarly to AR model, moving average model is also regression. Current value is a regression against white noise of prior values of the series. A random noise from each point is assumed to come from the same distribution which typically is a normal distribution.

Model of order q is written as

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_p y_{t-q} \tag{2.6}$$
$$e_t \overset{iid}{\sim} N(0, \sigma^2)$$

Moving average smoothing should not be confused with this model. Moving average smoothing is used for estimating a trend while this model is used for forecasting future values.

It is important to mention that any stationary $AR(P)$ model can be written as $MA(\infty)$

2.5.3 ARIMA

2.6 White noise

White noise is stationary, it looks the same at any period of time.

2.7 Stationarity

Non – stationary time series can be transformed to stationary by computing differences between consecutive observations. It eliminates trend and seasonality. For random walk next value can by written as $y_t = y_{t-1} + e_t$. Random walk can describe non – stationary time series. It has following characteristics:

- long periods of apparent trends up or down
- sudden and unpredictable changes in direction

Forecast for this model are equal to last observation(naive method) because future movements are unpredictable.

3 Evaluating Forecast Accuracy

In order to evaluate model it is important to estimate an error of the forecast. There are several methods for evaluating forecasting errors. Chosen were two MAE and RMSE. They are very similar however RMSE gives relatively high weight to larger errors.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

4 Used Forecasting Techniques

In the previous chapter several methods for forecasting were discussed, however in our system only a few of them were selected and implemented.

4.1 Metrics in Hawkular

In Hawkular there are three types of metrics: gauge, counter and availability. All of them are univariate metrics of structure $\{timestamp, value\}$.

5 Design and Implementation

5.1 Architecture

As was sad module was developed as module of Hawkular project, therefore was important to follow architecture of whole application. Alert prediction module was developed as standalone web application in Java language. Communication with other modules was accomplished through Java Message Service and REST calls.

Following diagram 5.1 shows how this module fits to Hawkular application. Almost all of the modules are developed separately and can work without each other. However this module is depended on metrics module and alerts.

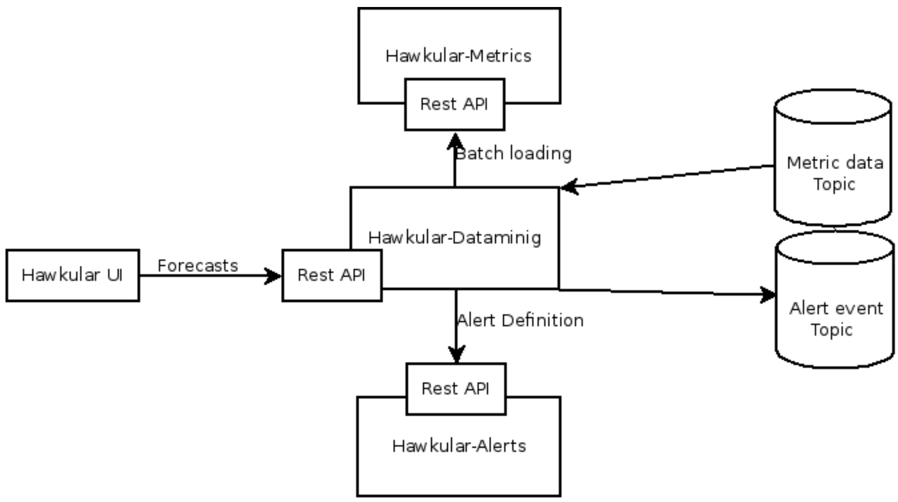

Figure 5.1: Architecture

6 Evaluation

7 Conclusion

[1]

Bibliography

[1] Shalunov, S.; Teitelbaum, B.; Karp, A.; aj.: A One – way Active Measurement Protocol (OWAMP). RFC 4656. Září 2006.