Exponential Moving Maximum Filter for Predictive Analytics in Network Reporting

Bin Yu, Les Smith, Mark Threefoot
Advanced Technology, CTO Office
Infoblox Inc.
Santa Clara, California, USA
e-mail: {biny,lsmith,mthreefoot}@infoblox.com

***Abstract*—In networking industry, there are various services that are mission critical. For example, DNS and DHCP are essential and are common network services for a variety of organizations. An appliance that provides these services comes with a reporting system to provide visual information about the system status, resource usage, performance metrics, and trends, etc. Furthermore, it is desirable and important to provide prediction against these metrics so that users can be well prepared for what is going to happen and prevent downtime. Among the predictive measures, there are multiple metrics to reflect peak or maximum values such as peak volume or resource usage in networking. The peak value prediction is critical for the IT managers to ensure its organization is ahead of the cycles in terms of the network capacity and disaster recovery. There have been many algorithms and methods for prediction of trended time series data. However, peak values often do not fall into a trend by nature. The traditional trend prediction methods do not perform well against this type of data. In this paper, we present a novel filtering algorithm named "Exponential Moving Maximum" (EMM), this filter is used before applying a prediction algorithm against peak time series data. We also provide some experimental results on real data as a comparison to show that the prediction method has better accuracy when EMM filtering is applied to certain categories of networking data.**

Keywords-predictive analytics; trend forecasting; networking reporting; time series data; sequential pattern mining

I. INTRODUCTION

There have been a number of methods that can do trend prediction or forecasting on time series data by removing so called non-stationarity or noise. Simple moving average is the most basic technique that averages the last n observations of a time series [1][2]. It is appropriate only for very short or irregular data sets, where features like trend and seasonality cannot be meaningfully determined, and where the mean changes slowly. Exponential smoothing, such as the Holt-Winters method, is a more complex moving average method that involves parameters reflecting the level, trend and seasonality of historical data. It usually gives more weight to recent data [2]-[6]. An even more complex class of moving average models, autoregressive moving average (ARMA) [2][6]-[8] is capable of reflecting autocorrelations inherent in data. It can out-perform exponential smoothing when the historical data period is long and data is nonvolatile. But it doesn't perform as well when the data is statistically messy. The typical application of this forecasting technique is in marketing for which J. Armstrong *et al.* had a review on many methods in their publication [9].

One of the most active research areas employing trend prediction is stock market forecasting. Therefore, many researchers have applied different analysis methods to do stock trend prediction, including associative rule based approaches, chart pattern recognition, template matching, neural networks and SVM [10][11]. K. Wu *et al.* recently presented a method to predict stock trend with k-means clustering algorithms [12] in identifying patterns within a sliding window. However the complexity of the algorithm poses a limitation for methods that are used in real time applications.

Time series data generated by a network service system such as DNS and DHCP servers often contains useful non-stationarity of which an example is illustrated in Figure 1 that is a time series DNS query data with hourly maximum or peak values for a period of 270 days. Users, typically from IT departments, are interested in seeing the trend of peak value data and, furthermore, to know the prediction for near future. Therefore, they can have means to assess the capacities of their network allowing purchase and deployment of new equipment to meet expected demand with out having to over provision.. When a traditional prediction algorithm is used with this data, the information about the local maximums will unfortunately get lost.

In Section II, we present the algorithm of exponential moving maximum and its memory complexity. Section III provides the command lines and workflow for the integration with Splunk [13]. We present the experimental results with comparison in Section IV. The conclusion is presented in Section V.

II. EXPONENTIAL MOVING MAXIMUM FILTER

The EMM filter is used to aggregate historical values with a maximum aggregator so that the effect of these values can be taken into account by subsequent values whilst applying a magnitude decaying exponential along with time. That's similar to one of the special cases in ARIMA that's exponential moving average [14]-[16] where historical values are aggregated into the following values but with an average aggregator. The EMM filter can be defined as

 ISBN: 978-1-61208-415-2

$$y_k = \max_{0 \le i \le k} \{\alpha^{\frac{i}{w}} x_{k-i}\}$$

where

$$\alpha \in [0,\ 1.0]$$

is an inheritance parameter and w is a filtering window size. In the case when

$$\alpha = 0, y_k = x_k$$

and when

$$\alpha = 1, y_k = \max_{0 \le i \le k} \{x_i\}$$

If

$$y_k = \alpha^{\frac{m}{w}} x_{k-m}$$

then x_{k-m} is called the bubble point of y_k and m is the bubble distance. Figure 2 illustrates the relationship of the parameters in which the original value x will have a contribution on the magnitude of αx for the filtered value at a future position that is w distance away from x. It shows the future impact of a value is decayed exponentially over time.

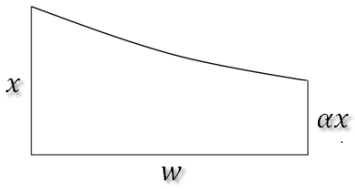

Figure 2. EMM filter parameters.

Figure 3 shows an example of EMM filtering over the hourly peak DNS query time series data. The peak values that are local maximums can become bubble points that over shadow the following non local maximal data points. The red curves show the EMM filtering output which effectively preserves the historical information of peak values and can contribute to the prediction of future peak values.

It can be proven that

$$\begin{aligned} y_k &= \max_{0 \le i \le k} \{\alpha^{\frac{i}{w}} x_{k-i}\} \\ &= \max\left(x_k, \alpha^{\frac{1}{w}} x_{k-1}, \alpha^{\frac{2}{w}} x_{k-2}, \dots, \alpha^{\frac{k}{w}} x_0\right) \\ &= \max\left[x_k, \alpha^{\frac{1}{w}}\left(x_{k-1}, \alpha^{\frac{1}{w}} x_{k-2}, \dots, \alpha^{\frac{k-1}{w}} x_0\right)\right] \\ &= \max\left(x_k, \alpha^{\frac{1}{w}} \max_{0 \le i \le k-1} \{\alpha^{\frac{i}{w}} x_{k-1-i}\}\right) \\ &= \max\left(x_k,\ \alpha^{\frac{1}{w}} y_{k-1}\right) \end{aligned}$$

This is the EMM representation in a recursive format that simplifies memory complexity to $O(1)$ for implementation.

III. SPLUNK CUSTOMIZATION

Splunk is a commercial software solution that provides archiving, indexing and analytics functions to machine generated data such as system logs and network data. One of its analytical functions is called *predict()* that can do forecasting based on a series of historical data points. As a platform, Splunk provides an SDK for users to develop custom commands as plug-ins. A custom command named *emm* is developed in Python and plugged into the Splunk system. The syntax of the command is as follows.

```
emm <variable_to_predict> [inheritance=i] [window=w]
```

where i is a floating point value between 0 and 1.0 to represent α and w is an integer value equal to the timespan. For instance, for hourly time series data, w=720 has a window size of one month.

In addition, the native Splunk command *predict()* is customized into a new command *forecast()* with following syntax.

```
forecast <variable_to_predict> [AS <newfield_name>]
[<forecast_option>]
```

The forecast_option is similar to the options for the command *predict()* except for some fields that have different default values customized.

With the custom commands deployed into Splunk app, the prediction process with EMM filtering can be pipelined similar to most of the Splunk queries. A sequence diagram is given in Figure 4 with the steps listed as follows.

1. Load event data files into Splunk system.
2. Fire a query to start the prediction process.
3. The query starts from event aggregation with use of Splunk aggregation functions.
4. Apply EMM filtering on aggregated time series data.
5. Further aggregate EMM output into a coarse level desired by prediction objectives.
6. Execute custom forecast command to get prediction result.
7. Visualize prediction result with Splunk visualization functions.

A sample query command pipe in Splunk is given as follows.

```
source="dns.txt" | rex
"^(?P<date>[^\\t]+)\\t(?P<dns>.+)" | timechart
span=10m max(dns) as dns | emm dns inheritance=0.7
window=4320| timechart span=mon max(emm) as emm
| forecast emm future_timespan=3
```

The output is shown in Figure 5 in which the top section is for command input and the lower part shows the EMM filtering and prediction results.

IV. EXPERIMENT AND COMPARISON

Infoblox is a company that provides DNS, DHCP and IP address management (DDI) appliances for network automation [17]. Its Trinzic DDI™ series is distributed with a Grid Master™ and a number of Grid Members™. The

reporting appliance can be one of the members that collects, archives and analyzes the network data across many members. About 18 month data is collected from two separate customers who are using Infoblox's Trinzic™ Reporting appliance. The data includes number of DNS queries aggregated every 10 minutes, number of DHCP leases aggregated every one minute, DNS server cache hit rate, and system CPU usage history. For each category, the data is segmented into a range of 12 months with a window sliding by month. To simplify the computation, EMM filter is applied on hourly maximum and the prediction is executed on monthly maximum data points. The experiment will use the first consecutive 9 month data from each segment to predict the values for the next three months. The prediction results will then be compared to the reserved three month data for accuracy calculation. The prediction error is defined as a mean squared error

$$MSE = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

where $\hat{Y}_i$ is the prediction value of Y_i at the $(i+9)$th month and $i = 1, 2, 3$. m is the number of sliding steps. Based on MSE, the comparison metric is defined as

$$C = \frac{MSE_{EMM}}{MSE}$$

where MSE_{EMM} is the prediction error with use of EMM filter and MSE is the prediction error without use of EMM. It is apparent that $C < 1$ means accuracy improvement.

First of all, use the Splunk built-in prediction function to analyze the data and provide prediction results in the protocol set above. The raw data is in system log format that is loaded and parsed by a custom regular expression to extract the values from raw events. The software does aggregation on event data to generate time series sequences before applying its *predict()* command. The prediction results can be visualized by its built-in GUI or exported into a text file and visualized separately as illustrated in Figure 6, where the prediction results are shown in the gray area. Secondly, we apply the EMM filtering before invoking the *predict()* command. The EMM filtering results are superimposed on to the raw DNS data that is shown in Figure 6 and highlighted in red in Figure 7. Its prediction results are illustrated in the gray area in Figure 7. Unlike the version without use of EMM in Figure 6 that shows a relatively flat trend, the version with use of EMM in Figure 7 effectively shows the upward trend that matches real data. The same experiments are conducted on all of four categories of network data. For comparison, the above defined C values are calculated and listed in Table 1. The C values for DNS query and DHCP lease data are much smaller than 1 which proves a significant improvement in prediction accuracy. On the other side, the prediction accuracy improvement on the DNS hit rate data is not significant and the accuracy decreases on the CPU data. We will try to provide some explanation in next section.

TABLE I. ACCURACY COMPARISON

Test Data	Comparison Metric C
DNS Query	0.07
DHCP Lease	0.33
DNS Hit Rate	0.98
CPU	1.21

V. CONCLUSION

Many prediction algorithms and methods have been experimented against the time series data that contains non-stationary peak values with poor performance. A preprocessing approach is proposed in this paper as well as an exponential moving maximum filter that can preserve local maximum values from the historical data and make prediction more accurate, meaningful and useful. An example EMM plug-in has been tested with use of Splunk software that provides an ease-of-use user interface and SDK for customization with plug-ins. The same method and algorithm can be used together with other prediction tools or software. The experimental results show that using the EMM filter provides better prediction accuracy on DNS and DHCP data compared to a traditional prediction algorithm provided by some commercial software. Unlike the experiments for DNS and DHCP volume data, the experiments on cache hit rate data and CPU data either show no improvement or present slightly worse accuracy. The possible explanation based on the sample data is that the spikes in cache hit and CPU data look more like real noise than trended peaks in DNS and DHCP volumes. This concludes that EMM should only be applied to the time series data that contains non-stationality which is intrinsically not random noise. Further experiments are needed to add EMM filter into other traditional predication algorithms and methods.

REFERENCE

[1] E. Booth, J. Mount, and J. Viers: "Hdrologic Variablility of the Cosumnes River Floodplain," San Francisco Estuary and Watershed Science, vol. 4(2), 2006.

[2] S. Makridakis, S. Wheelwright, and R. Hyndman, "Forecasting: Methods and Applications (3rd ed.)," New York: John Wiley & Sons, 1998.

[3] J. Armstrong, Principles of Forecasting: A Handbook for Researchers and Practitioners (Section 8: "Extrapolation of time-series and cross-sectional data"). Boston, MA: Kluwer Academic, 2001.

[4] E. Gardner, "Exponential Smoothing: the State of the Art," Journal of Forecasting, vol. 4, 1985, pp. 1-28.

[5] E. Gardner, "Exponential smoothing: The state of the art – Part II." International Journal of Forecasting, vol. 22, 2006, pp. 637-677.

 ISBN: 978-1-61208-415-2

[6] D. Montgomery, C. Jennings, and M. Kulahci, Introduction to Time Series Analysis and Forecasting. Hoboken, N.J.: 34.: Wiley-Interscience, 2008.
[7] G. Box, G. Jenkins, and G. Reinsel, "Time Series Analysis: Forecasting and Control," 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.
[8] S. Makridakis, M. Hibon, "ARMA Models and the Box–Jenkins Methodology". Applied Econometrics (Second ed.). Palgrave MacMillan, ISBN 978-0-230-27182-1, 2011, pp. 265–286.
[9] J. Armstrong, R. Brodie, and S. McIntyre, Forecasting Methods for Marketing: Review of Empirical Research, International Journal of Forecasting, Vol. 3, 1987, pp. 355-376.
[10] C. Slamka, B. Skiera, and M. Spann, "Prediction market performance and market liquidity: A comparison of automated market makers," IEEE Transactions on Engineering Management, vol. 60, 2013, pp. 169-185.
[11] A. Zhu and X. Yi, "The comparisons of four methods for financial forecast," in Proceedings of IEEE International Conference on Automation and Logistics, 2012, pp. 45-50.
[12] K. Wu , Y. Wu and H. Lee, "Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining," Journal of Information Science and Engineering, vol. 30, 2014, pp. 653-667.
[13] http://www.splunk.com, 2015.
[14] R. Brown, Exponential Smoothing for Predicting Demand, Cambridge, Massachusetts: Arthur D. Little Inc. 1956, pp. 15.
[15] J. Lucas and M. Saccucci, "Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements," Technometrics, vol. 32, 1990, pp. 1-29.
[16] C. Lowry, W. Woodall, C. Champ, and S. Rigdon, "A Multivariate Exponentially Weighted Moving Average Chart," Technometrics, vol. 34, 1992, pp. 46-53.
[17] http://www.infoblox.com, 2015.

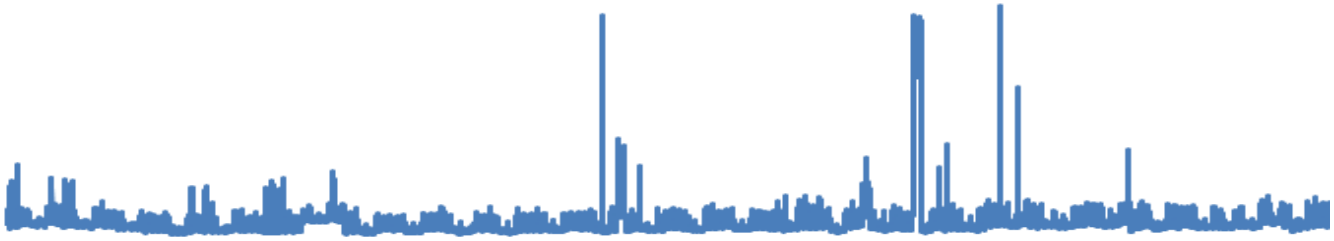

Figure 1. Customer DNS volume data example with meaningful non-stationality.

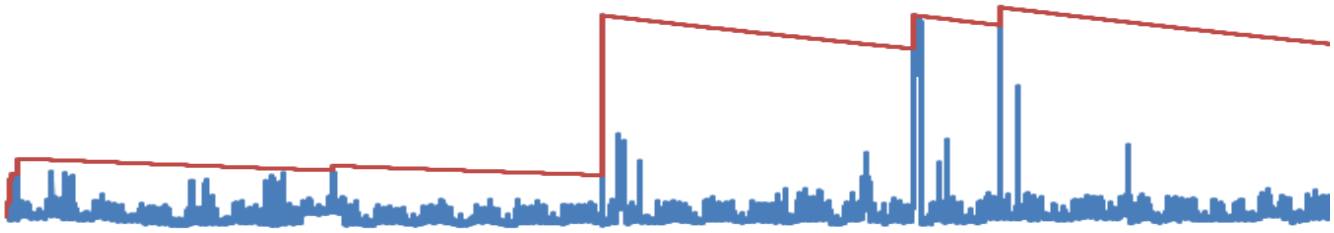

Figure 3. EMM filtering example.

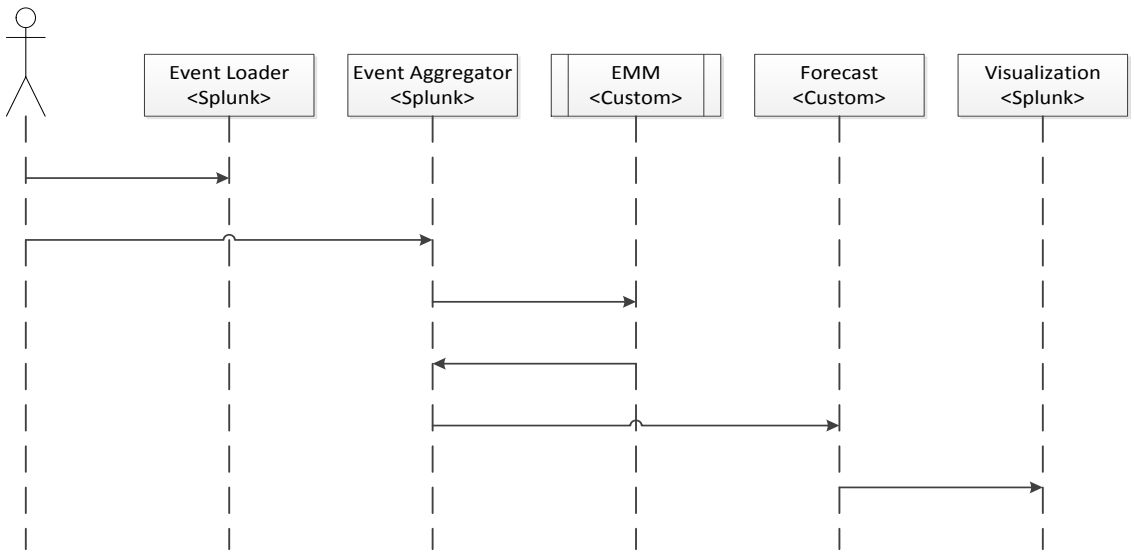

Figure 4. Sequence diagram of prediction process with EMM on Splunk.

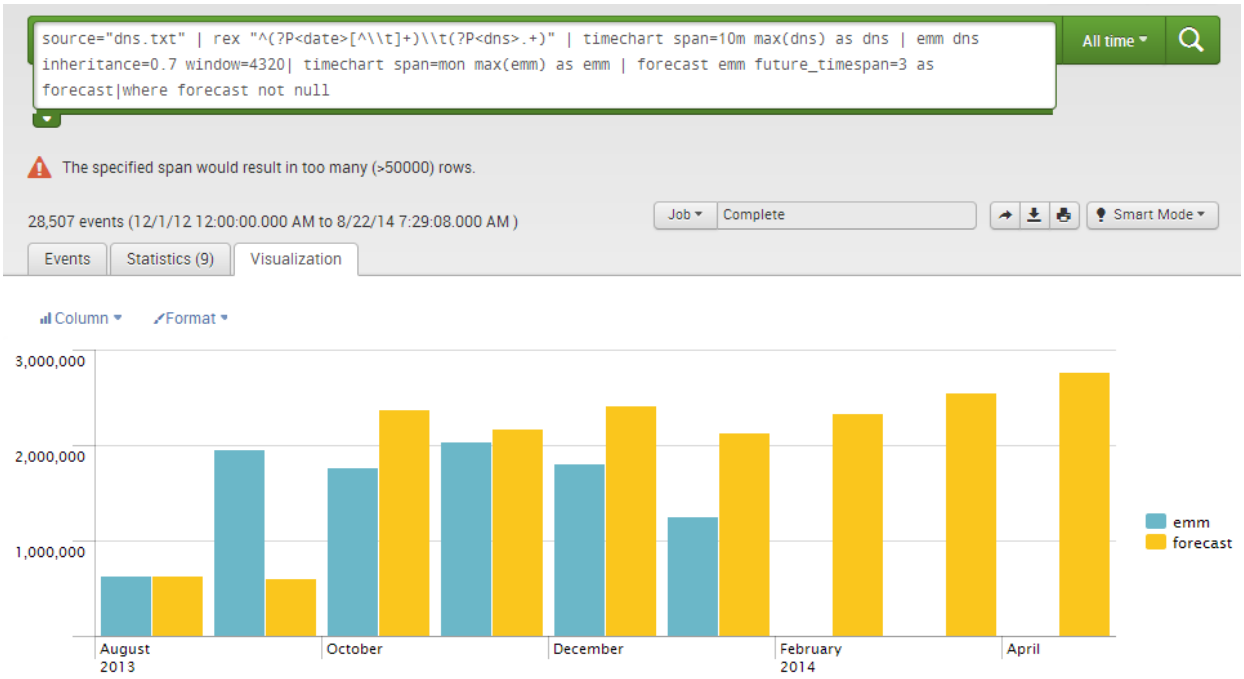

Figure 5. A screen snapshot of the web GUI of Splunk prediction with EMM plug-in.

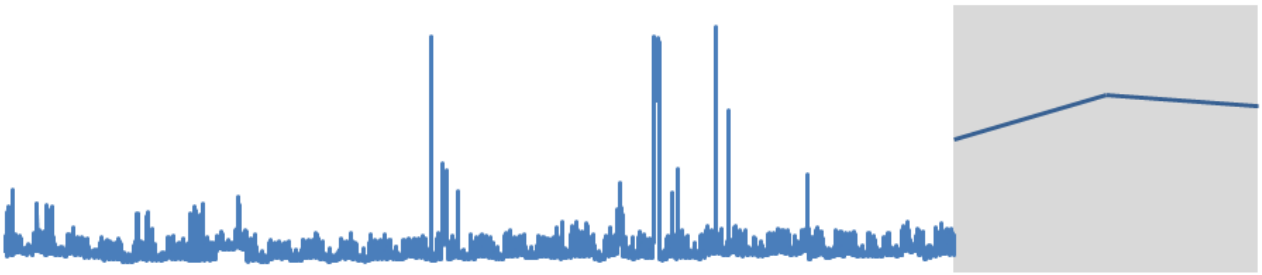

Figure 6. Prediction without use of EMM filter.

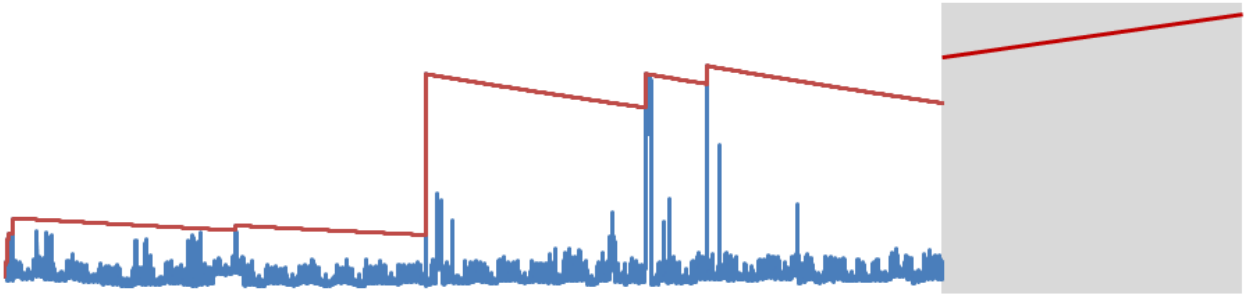

Figure 7. Prediction with use of EMM filter.

 ISBN: 978-1-61208-415-2