Cloud Service Modeling

Viveka Kulharia*
Microsoft Research India
t-vikulh@microsoft.com

Arushi Jain*
Microsoft Research India
t-arujai@microsoft.com

Shwetabh Khanduja*
Microsoft Research India
shkhandu@microsoft.com

Sundararajan Sellamanickam
Microsoft Research India
ssrajan@microsoft.com

ABSTRACT

Modeling of the multivariate time-series is a challenging but important problem given the various requirements which include predicting the future behavior of the time-series and anomaly detection. Here we model the real world multivariate time-series data such that we can use it to not only generate synthetic time-series with real world characteristics but to also create efficient and effective anomaly detection algorithm. This approach is better than the current models which are heuristics based and hence get troubled by joint modeling.

1. INTRODUCTION

The paper to be cited are [Killick et al., 2012], [Adams and MacKay, 2007] and [Xuan and Murphy, 2007].

Here we explore the use of change-point detection algorithms for anomaly detection such that they can be easily acted upon.

We found that real-world time-series can be easily categorized into four kinds of histograms. And with this modeling assumption, we created synthetic time-series as well as the

2. MOTIVATION

Application Centric The probabilistic modeling for data generation process - Generative modeling

After observing the real data, it was noticed that any real data time-series could be characterized by four different kind of distribution. The distributions follows four kinds of distribution: Left exponential, Right Exponential, U shaped and Inverted U shaped. These distributions are shown in figure . These distributions are modeled well using Beta distribution. Beta distribution is a family of continuous probability distributions which are defined on the interval [0, 1]. They are parameterized using two positive shape parameters, denoted by α and β, that are exponents of the random variable which controls the shape of the distribution. Real world time-series portrays the characteristics of levels of issues which signifies about the severity of any anomalous behavior. We have modeled four different kinds of behavior: normal time-series, low severity, medium severity and high severity issues. High severity issues occur less frequently than medium issue and similarly medium issue occur rarely that low severity issue in real world. Apart form that the distribution of time duration of a severity depends on the type of the severity. It is likely that high severity issues would occur for shorter duration as compared to medium or low severity issue as it would get resolved at faster pace due to its higher disruptive impact on services. Further, each time segment would also have a minimum length defined by a service, because each service takes certain amount of time to exhibit anomalous behavior.

We have generated a probabilistic model for characterizing the behavior of real multivariate time series. A typical service is instrumented by various sensors including counters which measures performance of the service in terms of latency, memory usage, disk usage, job success rate, etc. The service uses these counters to measure and diagnose the issues. We have characterized the overall heath of the service in terms of normal state, some known issue type and unknown issue type. The known issue type could contain several issues which would involve one or more counters. Same holds true for unknown issue type which could contain issue that is not known to data engineer or is occurring for the first time. The health state severity i.e. low, medium, high or normal is determined by the number of counters and their respective issue severity, involved in an issue. The type of the issue in a time series determine the characteristic of the data sample in time-series. When a high severity issue occurs in a service, it gets resolved earlier as compared to low or medium severity issue, thus guiding the duration of the erroneous behavior in a time-series. The following equation describes the modeling of the Cloud Service.

$$P(X, CTS, LCTS, IT, H) = P(X|LCTS, CTS) * P(CTS|H) * P(LCTS|IT, H) * P(IT|H) * P(H) \quad (1)$$

$P(cs_i^l|cs_{i-1}^l)$ Modeling

(2)

3. RELATED WORKS

*Equal Contribution

4. MODELS

How do we model this problem Probabilistic model Before describing our model we introduce the following notations. $L \in \{N, L, M, H\}$ denotes the health state of a counter/service with values representing normal state and abnormal states of low, medium and high severity levels. $D \in \{LE, RE, U, IU\}$ denotes the four types of distributions that each counter time series can follow. The values stand for left exponential, right exponential, U shape and Inverted-U shape. Refer to the motivation section for sample plots of each type $Is \in \{N, KIT_1, KIT_2, \cdots, KIT_n, UKIT\}$ denotes the type of issue occurring in the service for a given time segment. Its values denote either no issue, i^{th} known issue type or unknown issue type. $He \in \{N, KIT_H, KIT_M, KIT_L, UKIT_H, UKIT_M, UKIT_L\}$ denotes the overall severity of the service for a given time segment. It can be either normal or it could be having a known/unknown issue of low, medium or high severity. $CTS =< \tau_0^1, \cdots, \tau_{k_1+1}^1, \cdots, \tau_0^n, \cdots, \tau_{k_1+1}^n >$ denotes the segmentation for the counters where τ denotes a time instant. $GTS =< \tau_0, \tau_1, \cdots, \tau_k, \tau_{k+1} >$ denotes the segmentation for the service with k change-points. $H =< He_1, He_2, \cdots, He_{k+1} >$ denotes the health state of the service for each of the k+1 segments. $IT =< Is_1, Is_2, \cdots, Is_{k+1} >$ denotes the type of issue going on in the service doe each of the k+1 segments. $TT =< D_1, D_2, \cdots, D_n >$ denotes the time series types for all the counters. $LCTS =< L_1^1, \cdots, L_k^1, \cdots, L_1^n, \cdots, L_k^n >$ denotes the severities of different segments obtained after segmentation for all the counters. $LGTS =< L_1, L_2, \cdots, L_k >$ denotes the severities of the global segments achieved after combining the individual counter severities. $X_{n \times m}$ denotes the observed data time series for n counters each of length m.

We write the cloud service model as:

$$\begin{aligned} &P(H, IT, LGTS, GTS, LCTS, TT, CTS|X) \\ &\quad = P(H|IT, LGTS)P(IT|LGTS) \\ &\quad P(LGTS|GTS, LCTS)P(GTS|CTS) \\ &\quad P(LCTS|TT, CTS)P(TT|X)P(CTS|X) \end{aligned} \quad (3)$$

5. OUR SOLUTION

5.1 Segmentation

It is used to localize the anomalous behavior in a given input time series to exact segments within it. It outputs a list of segments. It works in two stages. In the first stage it segments the input time series by running a segmentation algorithm which identifies the change-points present in the series. In the second stage it scores each segment (i.e. interval within two change-points), to decide whether they are anomalous or not. This is because not all the change-points given by the first step are anomalous. Below is its further explanation:

1. Binary Segmentation: This is an O(nlogn) method that works by recursively finding the change-points in a given time segment [Scott and Knott, 1974], [Eckley et al., 2011]. Following is the criteria to detect whether there is a changepoint present or not: Given a segment $y_{(1:n)}$ a changepoint exists at $1 < \tau < n$ if $C(y_{1:\tau}) + C(y_{(\tau+1):n}) + \beta < C(y_{1:n})$ We have the following three variants of the method:
   (a) Linear Binary Segmentation: This method linearly scans over the given segment for a changepoint that satisfies the above condition and returns the very first changepoint.
   (b) Optimal Linear Binary Segmentation: This method linearly scans over the given segment and returns the changepoint that has the maximum gap between the LHS and RHS of the above equation
   (c) Mid Point Tester: This method treats only the middle point of the segment to be the candidate change point and tests whether that satisfies the above equation or not.
2. Optimal Segmentation : This is a $O(n^2)$ dynamic programming approach method that minimizes the below objective function[Eckley et al., 2011].
$$\min \Sigma_{i=1}^{m+1}[C(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(m) \quad (4)$$
where τ is start or end point of a segment, m is total segments, C is cost of a segment, $f(m)$ penalty as a function of total segment and β is relative importance of the penaltly with respect to Cost.
3. PELT: This is a method that improves on the running time of the above method by reducing the search space. Under certain assumptions this method given O(n) time complexity[Killick et al., 2012].
4. Vi Segmentation: This is O(n) time implementation.

The cost function used is: Negative Log Likelihood:

$$-\max_{\theta} \Sigma_{i=s}^{e} log f(y_i|\theta) \quad (5)$$

For discrete we have three variants:

1. $\Sigma_{i=1} logp(y_i)$
2. $logp(y_1) + \Sigma_{i=2} logp(y_i|y_{i-1})$
3. $logp(y_1) + \Sigma_{i=2} logP(y_i|y_{i-1})$ where $P(y_i|y_{i-1})$ is computed using the whole time-series.

We used two variants of penalty function β

1. AIC: 2p
2. BIC: plog(n)

where p is total number of parameters for a segment and n is the segment length.

5.2 Labeling

We tried two variants of labeling:

5.2.1 Clustering

For every segment obtained using segmentation, a window of length w was run while estimating the multinomial parameters for the same. Then clustering is done using k-means over the estimated parameters, to get four clusters. Then based on the size of each cluster, they are assigned the labels Normal, Low, Medium and High. Now, parameters are estimated for each of the segments and then according to the nearest cluster center, the respective segment is assigned a label.

5.2.2 Beta Param

For every segment obtained using segmentation, beta parameters are estimated by first estimating the multinomial parameters. The objective function minimized is

$$f(p, \alpha, \beta) = \Sigma_{i=1}^{N}(p_i - \int_{a_i}^{a_{i+1}} B(x; \alpha, \beta)dx)^2 \quad (6)$$

where α, $\beta > 0$ and N is the number of discrete characters, p_i is the multinomial parameter for the i^{th} character and

$$B(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1} \quad (7)$$

Now, to optimize it iteratively, we need its differential w.r.t α and β which come out to be

$$\frac{\partial f}{\partial \alpha} = \Sigma_{i=1}^{N} 2(p_i - \int_{a_i}^{a_{i+1}} B(x; \alpha, \beta)dx)$$
$$(-\int_{a_i}^{a_{i+1}} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}[log(x) - (\psi(\alpha) - \psi(\alpha+\beta)]dx) \quad (8)$$

and similarly

$$\frac{\partial f}{\partial \beta} = \Sigma_{i=1}^{N} 2(p_i - \int_{a_i}^{a_{i+1}} B(x; \alpha, \beta)dx)$$
$$(-\int_{a_i}^{a_{i+1}} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}[log(1-x) - (\psi(\beta) - \psi(\alpha+\beta))]dx) \quad (9)$$

We used sequential least square programming to optimize the objective function. QUADPACK library was used for numerical integration.

6. EXPERIMENTAL WORK

7. RESULTS

Discussion and Future work
Have a look at

8. CONCLUSION

9. REFERENCES

[Adams and MacKay, 2007] Adams, R. P. and MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

[Eckley et al., 2011] Eckley, I. A., Fearnhead, P., and Killick, R. (2011). Analysis of changepoint models. *Bayesian Time Series Models*, pages 205–224.

[Killick et al., 2012] Killick, R., Fearnhead, P., and Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

[Scott and Knott, 1974] Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512.

[Xuan and Murphy, 2007] Xuan, X. and Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055–1062. ACM.