Implementation of A Robust Universal Outlier Filter for Online Experimentation

Yan He
Oath Inc
701 1st Ave
Sunnyvale, CA 94089
heyan@oath.com

Miao Chen
Oath Inc
701 1st Ave
Sunnyvale, CA 94089
miaoc@oath.com

Matthew Dinh
Oath Inc
701 1st Ave
Sunnyvale, CA 94089
dinhm@oath.com

Nikhil Mishra
Oath Inc
701 1st Ave
Sunnyvale, CA 94089
ngmishra@oath.com

ABSTRACT

Online experimentation has been widely employed for evaluating new product features at internet companies. Existence of outliers poses a threat to the validity of experiment results. In this research, a mechanism-independent outlier filter is implemented in a high-throughput experiment measurement system. The impact is quantified to highlight the importance of such protection against the outliers for online experimentation. This paper demonstrates that this universal outlier filter is scalable, robust and able to cope with various types of metrics across products and devices.

Keywords

Outlier Detection; Outlier Filtering; Online Experimentation; Web Analytics; Implementation; Hurdle Model; Outward Testing Procedure; OTPSM

1. INTRODUCTION

Outliers arise from multiple sources including data error, web crawler, impression fraud and click fraud, among others [6] [18]. The mechanism of outlier generation and detection has been extensively studied in the literature [4][10]. At Oath, the majority of robotic traffic has been removed with an in-house traffic protection engine. In practice, we observe that a small fraction of outliers can escape from this traffic protection engine and complicate data analysis, particularly in online experimentation.

Online controlled experiments, also called A/B tests, have been widely employed for evaluating new product features at internet companies including Amazon, Facebook, Google, LinkedIn, Netflix, Bing and Yahoo, etc. [11][14][13][15]. Existence of outliers poses a threat to the validity of experimental results that is crucial to the product decision making process. Recent research demonstrated that even a small amount of outliers can cause erroneous analysis results and, in extreme cases, completely revert experimental results [16][9]. In a large-scale experimentation platform, a filter is required to remove remaining outliers and cope with various types of metrics across products and devices. It is also desired not to depend on any specific outlier generation mechanism.

This paper will discuss the implementation of a mechanism-independent filter that solves the outlier problem at Oath. The background information related to a recently published statistical method is firstly presented to set the context of this research [9]. We focus on the implementation of this filter in a large experimentation platform. The performance is evaluated with actual experiment data that demonstrates that this universal filter is independent of outlier generating mechanism, scalable and versatile for various types of metrics commonly seen in the Internet industry.

2. METHODOLOGY

A distribution-based outlier detection method was recently introduced to quantify the probability of a data point being an outlier [9]. In this framework, a parametric Hurdle model [12] is proposed to model the distribution of online count data. Simulation study has shown it is performing better than commonly used outlier methods with highly skewed online data, including normality based methods, non-parametric methods (Boxplot and Hampel), and classification methods (Kmean and Hierachical clustering). It is a two-component model with a hurdle component to handle zero versus positive observations and a truncated Negative Binomial distribution to fit positive counts.

$$Y_i \sim \begin{cases} 0 & \text{with probability } 1-p \\ ZTNB(\rho,\gamma) & \text{with probability } p \end{cases} \quad (1)$$

where $ZTNB(\rho,\gamma)$ is the zero-truncated Negative Binomial distribution with Probability function

$$P(y_i|y_i>0,\rho,\gamma) = \frac{1}{1-(1-\rho)^\gamma}\frac{\Gamma(y_i+\gamma)}{y_i!\,\Gamma(\gamma)}(1-\rho)^\gamma\rho^{y_i}$$

and $\Gamma(\cdot)$ is the Gamma function.

AIC (Akaike Information Criterion) [2] is calculated with sample page view data that compares model fitting between Normal, Poisson, Negative Binomial and Hurdle model in Table 1. Hurdle model has the lowest AIC value which shows its superiority over the other three. Figure 1 illustrates the probability distributions of the four candidate models com-

Table 1: AIC of four models for example page view data

Model	AIC
Normal	613,363
Poisson	2,102,440
Negative Binomial	404,037
Hurdle Model	365,531

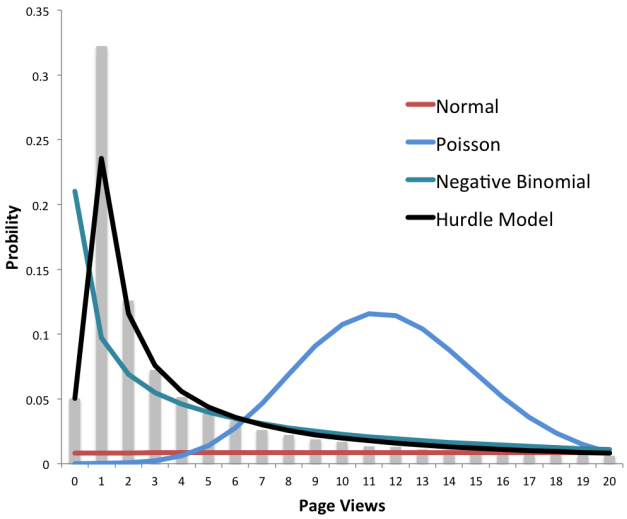

Figure 1: Probability estimation of four candidate models for example page view data

pared with the empirical distribution shown in the barplot. The page view is truncated at 20 in the figure to show the model fitting clearly. It is demonstrated that the Hurdel model outperforms all the other distribution models.

The observed sample maximum is subject to an outlier test based on the distribution of sample maximum derived from the Hurdle model. In order to handle masking and swamping effects as a result of multiple outliers [1][8], an outward testing procedure with sample maximum, also called OTPSM, is adopted as follows:

1. Calculate DFBETA[7] for each data point in the complete sample Y.

2. Divide Y into two sets Y_N and Y_O, where $Y_N = \{y_i : y_i \in Y \text{ and } DFBETA_i \leq \frac{2}{\sqrt{n}}\}$ and $Y_O = \{y_i : y_i \in Y \text{ and } DFBETA_i > \frac{2}{\sqrt{n}}\}$. Y_O contains all influential data points to be further examined.

3. Estimate the parameters of target distribution F using sample Y_N. For count data, we use the Hurdle model as described.

4. Locate the minimum value in Y_O, and test whether it is an outlier in Y_N with hypothesis testing.

5. If it is not an outlier, move it from Y_O to Y_N and iterate steps 3-5. If it is an outlier, the procedure terminates. The sample minimum of the resultant Y_O is set as the threshold for outliers.

3. IMPLEMENTATION

At Oath, an internal platform has been built for online experimentation across multiple web and app products in different countries, languages and devices [16][17][3][5]. Key metrics of each experiment, such as user actions and revenue metrics, are aggregated every day for statistical analysis. An outlier filter determined by OTPSM is added to the data pipeline before the results are presented in a Tableau dashboard to end users.

3.1 ETL Process for Experimentation Data

Figure 2 illustrates the architectural design of the implementation. When an experiment is created and deployed, a traffic splitter will start to randomly assign users into test samples with pre-defined size. Besides, the metadata of test samples flows into a dimension table that are used in the ETL process to identify test samples in a batch data pipeline.

Experiment data is collected every hour and aggregated at the user level every day for further processing. The daily metrics are grouped by product dimensions including product name and device, etc. For example, the page view events logged in Product C App are aggregated to the count of page views per user per day for the product.

3.2 Implementation of OTPSM

The proposed outlier filter is applied after daily metrics are aggregated. One challenge for implementing OTPSM is the burden of computation associated with determining each outlier threshold. It is an iterative procedure that consumes time and resource to run which is cost prohibitive to repeat for every single experiment metric when there are hundreds of experiments in our platform running in parallel every day. Given the fact that the distribution of normal data is generally stable for a product, we implement a universal and constant filter at the product level. That greatly reduces the number of runs of the algorithm without repeated computation for every single experiment. As an example, for Product C App we choose a commonly used metric such as page view and run the OTPSM procedure once to obtain the corresponding outlier threshold. Then it is applied in the data pipeline to remove the outliers with page views larger than this threshold. This approach allows us to implement a robust filter that may scale to power hundreds of concurrent experiments in our platform. Its performance will be evaluated in the following section.

4. PERFORMANCE AND EVALUATION

In this section we will discuss the performance of OTPSM outlier filter towards three commonly used online metrics: sessions, page views and clicks. The evaluation was done across six different online products at Oath: Products A, B and C in their desktop web and mobile app versions.

4.1 Robustness of Hurdle Model

In order for us to evaluate how well the underlying Hurdle model fits the online data, Table 2 quantifies the AIC of Hurdle model in comparison with other three conventionally used distributions including Normal, Poisson and Negative Binomial. The column 'Next Best' is the lowest (best) AIC among the other three competing distributions. Similar to what was observed in Section 2, the Hurdle model outperforms all the others consistently regardless metrics, products

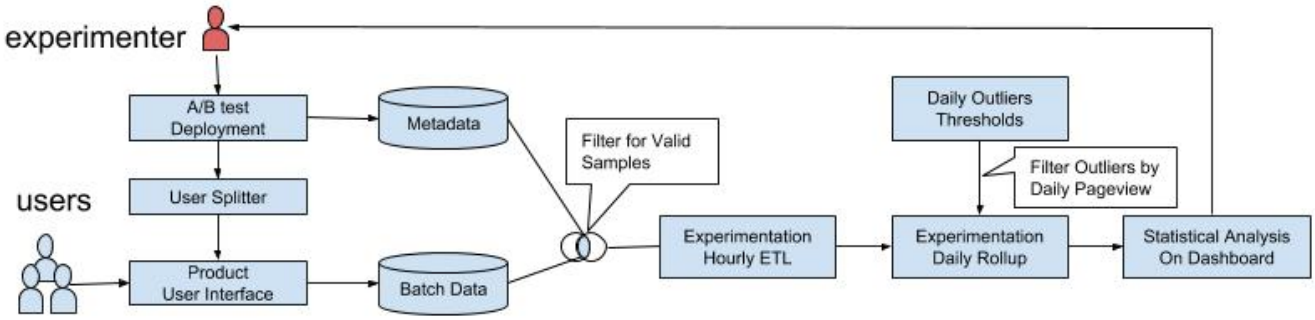

Figure 2: Implementation of OTPSM for Online Experimentation Platform

Table 2: Comparison of model fitting across multiple metrics and products

Metric	Product	AIC	
		Next Best	Hurdel
session	A Desktop	1,898,390	1,787,923
	A App	1,040,646	990,718
	B Desktop	1,737,177	1,539,834
	B App	3,779,008	2,900,204
	C Desktop	1,543,375	1,268,668
	C App	4,017,630	3,070,545
page view	A Desktop	3,740,012	2,536,109
	A App	3,261,149	2,271,675
	B Desktop	4,641,042	2,493,218
	B App	7,233,010	5,513,590
	C Desktop	5,539,038	2,609,565
	C App	8,599,680	6,733,096
click	A Desktop	4,008,392	2,495,492
	A App	2,297,250	1,487,001
	B Desktop	5,189,249	1,949,871
	B App	5,943,425	1,313,413
	C Desktop	8,599,680	6,733,096
	C App	4,886,767	545,335

or devices. Its flexible model structure enables itself to adapt to different distributions of online data, with improved AIC. This data demonstrated that the Hurdle model is versatile, robust and able to cope with various product settings.

4.2 Stability of Outlier Thresholds

As discussed in Section 3.2, the OTPSM filter is implemented at the product level to reduce the algorithm runs. Furthermore, the outlier threshold does not need to be updated frequently because it does not change much for the products with stable user base and usage patterns. As shown in Figure 3, we have examined the stability of outlier thresholds over a period of several months using both session and page view metrics. For Product A desktop, the page view outlier threshold (in blue, masked scale) varies within ±5% from late January to early May in the year of 2018. The variation for session outlier threshold (in red, masked scale) is even smaller.

4.3 Improvement of Experiment Measurement

The OTPSM filter improves the quality of experiment measurement in two folds. First, it removes the bias caused by outliers and improves the accuracy of experiment effect

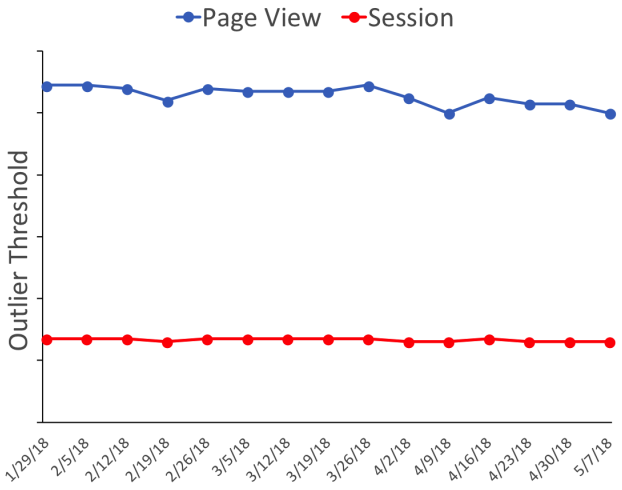

Figure 3: Outlier Threshold (Masked Scale) Over Time for Product A Desktop

measurement. Table 3 compares the experimental results before and after outlier filtering. If outliers are not properly removed, significantly different results can be concluded in terms of both magnitude and direction.

On the other hand, the OTPSM outlier filter decreases the metric variance and increases the statistical power of an experiment. As a result, it reduces the requirement of minimum sample size. In practice this benefit is strongly desired when sample size is a constraint, particularly for mobile apps with small user reach. Table 4 compares the required sample size before and after removing outliers with OTPSM, under the conditions of measuring 5% change in the metric of in-

Table 3: Correction of Experiment Effect Measurement with OTPSM outlier filtering

Test	Metric	Product	Experiment Effect	
			Before	After
1	session	A Desktop	.14%	.05%
2	session	B Desktop	-.98%	-1.5%
3	page view	A Desktop	.81%	.59%
4	page view	B Desktop	-2.3%	-2.8%
5	click	A Desktop	.66%	-1.0%
6	click	C App	-2.4%	-1.9%

Table 4: Reduction of sample size requirement with OTPSM outlier filtering

Metric	Product	Required Size		Reduction
		Before	After	
session	A Desktop	1605	1523	5%
	A App	3046	2651	13%
	B Desktop	1631	1585	3%
	B App	6721	5237	22%
	C Desktop	1636	1556	5%
	C App	7696	6647	14%
page view	A Desktop	15074	9637	36%
	A App	85732	12617	85%
	B Desktop	48654	24353	50%
	B App	30289	12573	59%
	C Desktop	58336	44238	24%
	C App	30409	16126	47%
click	A Desktop	17455	15768	10%
	A App	11603	9032	22%
	B Desktop	105369	42361	60%
	B App	247572	150232	39%
	C Desktop	54877	37981	31%
	C App	998714	151485	85%

terest with false positive rate $\alpha = 0.05$ and false negative rate $\beta = 0.2$. Due to the high variance caused by outliers, the required sample size could be as large as nearly a million, e.g. $998,714$ for measuring clicks in Product C App before outlier filtering. After the outliers are removed, the required sample size is reduced by 85% to $151,485$.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the implementation of a robust OTPSM outlier filter for online experiments. OTPSM employs a sequential testing procedure that is based on a universal Hurdle model. It does not rely on the learning of the underlying mechanism of outlier generation and can be applied in a broad spectrum of online data analysis.

The performance of OTPSM outlier filter was evaluated in a large online experimentation platform at Oath. The outlier threshold determined by OTPSM is consistent over time. The results from three metrics and six different online products across desktop and mobile devices demonstrate that the OTPSM filter can effectively correct the bias and reduce the noise caused by outliers in online experiment measurement. The OTPSM outlier filter is scalable, robust and able to function effectively at various product settings.

We demonstrate in Section 4.2 that the outlier threshold determined by OTPSM is sufficiently stable during a time period of several months. This allows us to skip the repeated threshold computation and simplify the implementation. In future research, it is desired to find a computationally economical solution for frequent threshold calculation at real time. Finally, the expansion from this univariate distribution-based filter to a multivariate distribution-based one, or at minimum expansion to a set of outlier thresholds for a collection of key metrics, should further improve the accuracy in online data analysis.

6. ACKNOWLEDGMENTS

We would love to extend our thanks to Sameer Raheja, Don Matheson, Maria Stone and Zhenyu Zhao for their valuable discussion along this project, and to William Choi, Charles Hartel, Asad Sheth and David Natali for setting up various experiments from which the data was collected for this research.

7. REFERENCES

[1] E. Acuna and C. Rodriguez. A meta analysis study of outlier detection methods in classification. *Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez*, 2004.

[2] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[3] N. Appiktala, M. Chen, M. Natkovich, and J. Walters. Demystifying dark matter for online experimentation. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 1620–1626. IEEE, 2017.

[4] A. Beutel, L. Akoglu, and C. Faloutsos. Graph-based user behavior modeling: from prediction to fraud detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2309–2310. ACM, 2015.

[5] R. Chen, M. Chen, M. R. Jadav, J. Bae, and D. Matheson. Faster online experimentation by eliminating traditional a/a validation. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 1635–1641. IEEE, 2017.

[6] N. Daswani, C. Mysen, V. Rao, S. Weis, K. Gharachorloo, and S. Ghosemajumder. Online advertising fraud. *Crimeware: understanding new attacks and defenses*, 40(2):1–28, 2008.

[7] B. S. Everitt. *The Cambridge dictionary of statistics*. Cambridge University Press, 2006.

[8] A. S. Hadi and J. S. Simonoff. Procedures for the identification of multiple outliers in linear models. *Journal of the American Statistical Association*, 88(424):1264–1272, 1993.

[9] Y. He and M. Chen. A probabilistic, mechanism-indepedent outlier detection method for online experimentation. In *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*, pages 640–647. IEEE, 2017.

[10] R. Kannan, H. Woo, C. C. Aggarwal, and H. Park. Outlier detection for text data. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 489–497. SIAM, 2017.

[11] R. Kohavi, A. Deng, R. Longbotham, and Y. Xu. Seven rules of thumb for web site experimenters. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1857–1866. ACM, 2014.

[12] J. Mullahy. Specification and testing of some modified count data models. *Journal of econometrics*, 33(3):341–365, 1986.

[13] C. Smallwood. The quest for the optimal experiment. *RecSys'14 Workshop: Controlled Experimentation*, 2014.

[14] D. Tang. Experimentation at google. *RecSys'14 Workshop: Controlled Experimentation*, 2014.

[15] Y. Xu and N. Chen. Evaluating mobile apps with a/b and quasi a/b tests. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining, pages 313–322. ACM, 2016.

[16] Z. Zhao, M. Chen, D. Matheson, and M. Stone. Online experimentation diagnosis and troubleshooting beyond aa validation. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 498–507. IEEE, 2016.

[17] Z. Zhao, Y. He, and M. Chen. Inform product change through experimentation with data-driven behavioral segmentation. In *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*, pages 69–78. IEEE, 2017.

[18] R. K. Zwicky. Click fraud detection, Feb. 2 2010. US Patent 7,657,626.