Making Online BigData Small: Reducing Computation Cost and Latency in Web Analytics through Sampling

Ariyam Das
Yahoo! Research and Development
Bangalore, India
ariyam@yahoo-inc.com

Harish Ranganath
Yahoo! Research and Development
Bangalore, India
harishsr@yahoo-inc.com

ABSTRACT

In the era of big data, the volume, velocity and variety of data are exploding at an unprecedented pace. With the explosion of data at the web, internet companies are working towards building powerful data analytics system that can crunch the big user data available to them and offer rich business insights. However, generating analytical reports and insights from high dimensional online user data involves significant computation cost. In addition, enterprises are now also looking for low latency analytics systems to dramatically accelerate the time from data to decision by minimizing the delay between the transaction and decision. In this work, we attempt to create smaller samples from the actual online big user data, so that analytical reports can be derived from the sample data faster and at a reduced computation cost without significant trade-offs in precision and accuracy. This study empirically analyzes petabytes of traffic data across Yahoo sites and develops efficient sampling and metric computation mechanisms for large scale web traffic. We generated analytical reports containing traffic and user engagement metrics from both sampled and actual Yahoo web data and compared them in terms of latency, computation cost and accuracy to show the effectiveness of our approach.

Keywords

Big data, web analytics, high dimensional data, additive metrics, non-additive metrics

1. INTRODUCTION

Web data analytics have become significantly relevant as internet companies begin to recognize the value of data-driven decision making [1]. Organizations have started collecting, storing and analyzing more granular information about their products, users and transactions than ever before. Mining this big data can provide multi-disciplinary perspectives and valuable information like user engagement patterns, consumer preferences and habits and so on. In web analytics, the server logs and page tagging techniques in web analytics [2] can capture user actions and behavior at the most granular level. These unstructured terabytes or even petabytes of data are converted to structured data, spanning across few hundreds or thousands of dimensions. Several OLAP applications [3] process this web data and publish canned reports and dashboards to business analysts for extracting meaningful insights. However, with the big data explosion, these applications not only have to process this enormous data, but also have to accommodate hundreds or even thousands of data dimensions.

The growing number of combinations of different dimensional values (cells) poses a huge challenge for the OLAP databases. As the number of dimensions increases, the number of cells for OLAP databases increases exponentially. Typically a 16 dimension database, with 5 members in each, requires 152 billion cells. Therefore, instead of accommodating the entire high dimensional data in a relational database, the usual approach taken is to group the raw data by selected combinations of dimensions and then to aggregate over it. The aggregated data for each of the dimensional combinations is then loaded into relational databases. Complex SQL queries are executed over these databases to generate different analytical reports consumed by business analysts to gain an overall view of the business health. With increase in cardinality of the user data, the number of different dimensional combinations for aggregating the raw data also drastically increases. Thus the growing size and dimensionality of the data consequently increases the computation overhead and latency of the analytical reports.

In this study we try to deal with these challenges by creating a smaller but an efficient sample from the raw big data that would facilitate generation of low latency analytic reports from the smaller sampled data at a reduced computation cost but within a desirable precision. As Laptev et al. had indicated in [4], that obtaining approximate results from sampled data is often the only way in which advanced analytical applications working on very massive data sets can satisfy their time and resource constraints.

The rest of the article is organized as follows. In section 2, we empirically analyze online web traffic across Yahoo sites and attempt to develop an efficient sampling mechanism and corresponding metric computation methodologies. The results of our proposed approach are evaluated in section 3. Finally, section 4 concludes the article.

2. EMPIRICAL ANALYSIS

Internet companies mostly instrument their web pages to

The 19th International Conference on Management of Data (COMAD),
19th-21st Dec, 2013 at Ahmedabad, India.

log granular details like module and link level events, user actions, user agent strings and so on. All these events are recorded in the logs with a timestamp. This unstructured raw data is afterwards transformed into structured high dimensional data. The latter is then processed and aggregated across different dimensional combinations and then finally loaded into databases. Both additive and non-additive metrics are required to be computed during the data aggregation. Additive metrics like page views, clicks (traffic metrics) or time spent (user engagement metric) can be rolled up across a dimension by simply adding across all the dimensional values. On the other hand, non-additive metrics, which mainly include unique count computation, like unique number of visitors, unique browser cookies or unique logged-in users, cannot be rolled up like additives.

For all our experiments and analysis, we use the web traffic data across popular Yahoo sites like Mail, Sports, Finance, Answers and so on. Usually the web traffic and user engagement pattern vary across the hours in a day. However, for a small period of time, the web traffic is more or less similar during that period. So, intuitively we split the entire web traffic into small time periods. Within one such time period, we consider a smaller time frame and expect that the web traffic occurring in this window will be almost similar across the entire time period. We apply this time-based sampling technique on the entire data collected to create an appropriate sample. This methodology should apparently work better than random sampling, since online web traffic varies over time. However, those dimensional combinations that contribute a negligible percentage in the overall traffic ($<< 1\%$) may get missed in the sampled data. But this can be ignored since business analysts are mainly interested in knowing the *top k* dimensional combinations ($k < 20$) that contribute to the bulk of the traffic ($> 90\%$). Another drawback of time-based sampling is that, there is a high probability of missing out on critical events like SLA violations, outages and high severity incidents. But here, we are mainly interested with reporting of traffic metrics like page views, clicks and user engagement metrics like time spent, number of user sessions; we do not focus on incident and outage reporting. In the next two subsections we discuss extrapolation techniques for additive and non-additive metrics, so that the absolute deviation of extrapolated approximate values from the actual metric values are within reasonable bounds.

2.1 Additive Metric Computation

For the time-based sampling, we divided the online web traffic for one day into intervals of 60 minutes. Within this 60 minutes period, we considered a window of 3 minutes (approximately 5% sample). With this sampled data, we computed the page views (additive metric) for several dimensional combinations. We then calculated the ratio of the total page views obtained from the actual data to the page views obtained from the sampled data for all the combinations of the dimensional values. This ratio is the actual extrapolation factor for the page views. Table 1 shows the extrapolation factor for some of the popular user operating systems and browsers. As observed from table 1, the page views across attribute values for different dimensions (like operating system and browser) have similar extrapolation factor (∼20). The same trend follows for all the dimensional combinations and for other additive metrics like clicks and time spent. Therefore, the extrapolation for additive metrics is almost linear (5% time-based sample having extrapolation factor of 20).

Table 1: Comparison of Extrapolation Factors for Page Views across Different Dimensions and Attribute Values

Dimension	Attribute Value	Extrapolation Factor
OS	Microsoft Windows	19.76
OS	Android	20.14
OS	Apple iOS	19.93
Browser	Internet Explorer	19.74
Browser	Firefox	20.04
Browser	Safari	19.88

We now the compare the performance of random sampling with our time-based sampling using 60 minutes time intervals. For varying sample sizes, we compared the absolute percentage error between the two sampling techniques. The results are captured in figure 1. In time-based sampling, the sample size is actually determined by the size of the sampling window in the considered time interval. As shown in figure 1, time-based sampling performs better than random sampling although the error percentages for both the sampling techniques gradually tend to converge as the sample size increases.

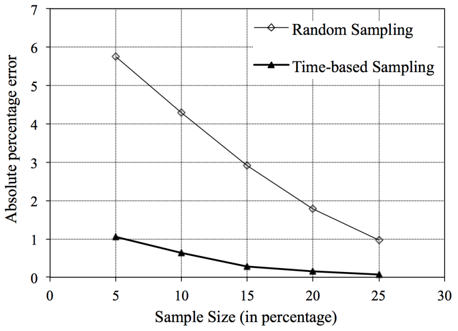

Figure 1: Performance comparison of time-based sampling with random sampling for additive metrics.

Figure 2 shows the impact of the length of the time period on the error percentage, considering a window size of 5% for all the time periods. As shown in figure 2, the absolute percentage error increases significantly as the length of the time period increases. Thus the time interval should be selected depending on the dynamic the nature of the traffic. Likewise, the time window within an interval should be decided based on the desired accuracy. From these experiments, we identified that a 5% window size over a time period of 30 minutes yields an error of less than 1% for additive metrics.

2.2 Non-Additive Metric Computation

In this section we will develop the extrapolation technique for non-additive metrics. Table 2 shows the extrapolation factor for the number of unique logged-in users (non-additive metric) for the same attribute and dimensional values shown in table 1, using the same parameters of the experiment. Unlike additive metrics, the extrapolation factors for non-

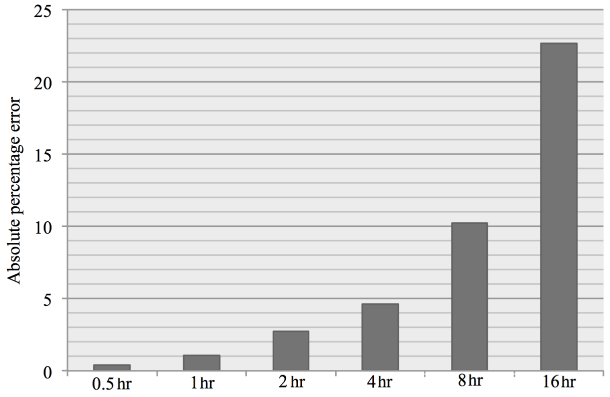

Figure 2: Impact of time period on absolute error percentage.

Table 2: Comparison of Extrapolation Factors for Unique Logged-in Users across Different Dimensions and Attribute Values

Dimension	Attribute Value	Extrapolation Factor
OS	Microsoft Windows	3.48
OS	Android	5.06
OS	Apple iOS	4.21
Browser	Internet Explorer	3.46
Browser	Firefox	5.10
Browser	Safari	4.02

additive metric are not uniform and these metrics cannot simply be linearly extrapolated. Hence for unique counts we need to follow a different approach.

Interestingly, it was observed for non-additives that the extrapolation factor for a combination of different dimensional values is generally more close towards the highest extrapolation factor of its constituent dimensions. For example, when the extrapolation factors of number of unique logged-in users for *Apple iOS* and *Firefox* are 4.21 and 5.1 respectively, the extrapolation factor for the combination of *Apple iOS* and *Firefox* is calculated as 4.97, which is closer to 5.1. Intuitively, we can expect that the dimensional value which has the highest extrapolation factor is not properly sampled, eventually leading to its high extrapolation factor. Therefore, any combination of dimensional values involving the former should be found under represented in the sample as well and hence is as less likely to appear in the sample. The same trend follows for other non-additive metrics as well. Thus, often the extrapolation factor for non-additives for a combination of different dimensional values tends to be closer to the highest extrapolation factor of its constituent dimensional values.

In order to empirically establish the accuracy of this argument, we calculate the non-additive metrics from the actual data for each of the dimensions and its attribute values taking one at a time. We also compute the non-additive metrics from the sample data for each of the attribute values of every dimension and calculate their respective extrapolation factors. For the different combinations of dimensional values, we calculate the non-additive metrics from the sample data and select the maximum extrapolation factor from the constituent dimensional values.

We follow this method and compare the performance of random sampling with our time-based sampling using 30 minutes time period. For a fixed combination of dimensional values, we vary the sample sizes and compare the absolute percentage error between the two sampling techniques. As shown in figure 3, our time-based sampling outperforms random sampling.

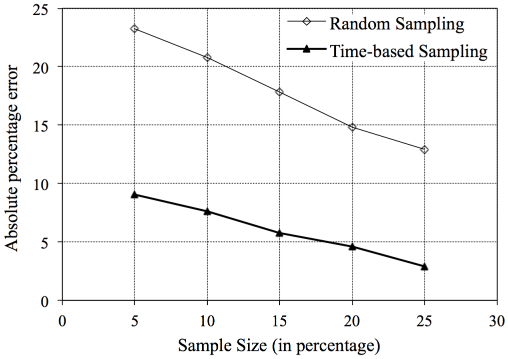

Figure 3: Performance comparison of time-based sampling with random sampling for non-additive metrics.

It is worth mentioning that a weighted mean of absolute percentage error (weighted absolute percentage error) is an appropriate measure to calculate the accuracy of our sampling technique; absolute percentage errors are calculated for each combination of dimensional values and the percentage contribution of the metrics are treated as their corresponding weights.

3. RESULTS

In this section, we present the results to assess the performance of our proposed approach. First we identify and capture the accuracy of the sampling mechanism. We generated some aggregated feeds from both the sampled and actual data. Due to memory limitations of relational databases, each aggregated feed usually contains 10 or fewer dimensions. We varied the number of dimensions and calculated the weighted absolute percentage error for additive (page views) and non-additive (count of unique logged-in users) metrics. The results are reported in figure 4 for additive metrics and in figure 5 for non-additives. The experiment was conducted taking the window size as 20% and time period as 30 minutes. As shown in figure 4, weighted absolute percentage error for additive metrics is approximately close to 0.14% across the number of dimensions. On the other hand, for non-additive metrics, the error percentage rises almost linearly as the number of dimensions increases. However, even for 10 dimensions, the error is less than 5%.

The two main merits of sampling is reduction of data footprint and faster processing. Due to our time-based sampling, usual backend ETA (extraction, transformation and aggregation) frameworks will work with considerably smaller data sets. Figure 6 compares the ratio of number of records in the sample to the number of records in the actual data for different window sizes. The linear graph in figure 6 implies that the window size in our time-based sampling is actually

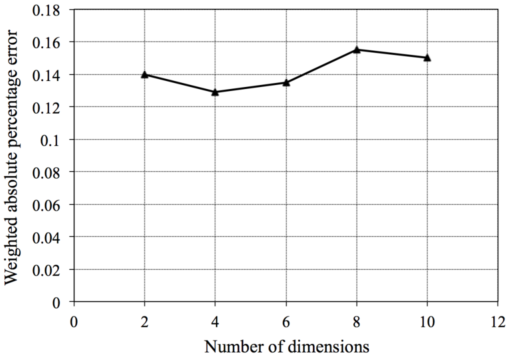

Figure 4: Weighted absolute percentage error for different number of dimensions for additive metrics.

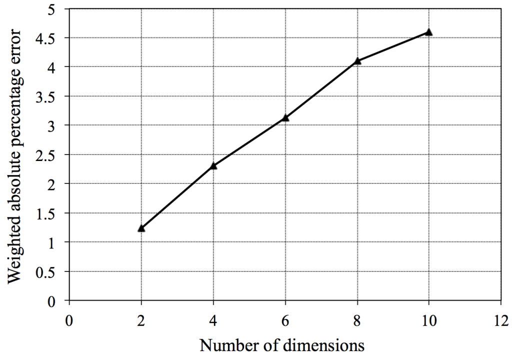

Figure 5: Weighted absolute percentage error for different number of dimensions for non-additive metrics.

equivalent to the sample size.

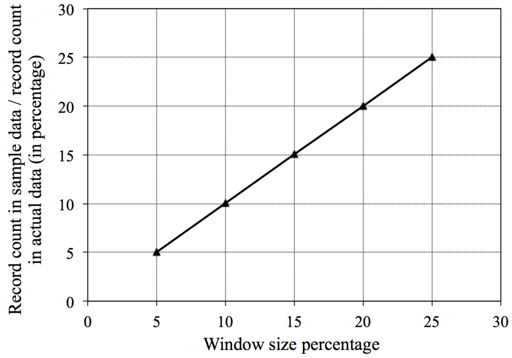

Figure 6: Relation between sample size and window size in our time-based sampling.

With reduction in data sets, the ETA framework processes the data faster and reduces the latency of analytical reports. We next determined the ratio by which the processing time has improved and how it varied as the number of dimensional combinations (group-by) increased. The experiment was conducted with the window size of 20% and time period of 30 minutes. It is evident from figure 7 that the processing time has approximately reduced to $\frac{1}{5}$ through our time-based sampling.

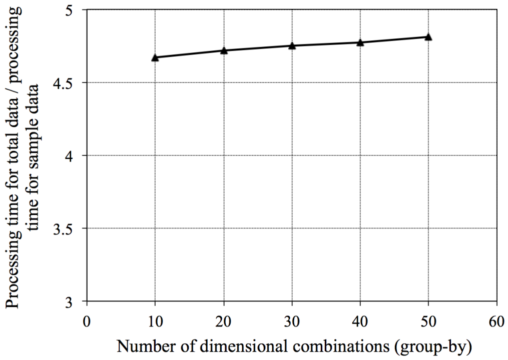

Figure 7: Improvement in processing time through time-based sampling.

All the above results show similar trend across several days. This shows that our sampling and metric computation mechanisms are stable and independent of the changes in the web traffic occurring over a week or month. In addition, our results obtained from different samples of the same data (with same sample sizes) are consistent with each other. This indicates that our sampling methodology has excellent precision as well.

4. CONCLUSIONS

The initial results of our time-based sampling method are encouraging. With this approach, we could calculate additive metrics (accuracy of < 1%) and non-additive metrics (accuracy of < 5%) with a reasonable accuracy as desired by the analysts. This sampling mechanism also greatly reduces the data footprint and latency of the analytical reports. Our future work is to improve the accuracy of non-additive metric computations through count sketch algorithms.

5. ACKNOWLEDGMENTS

The authors would like to thank Debasish Das, Rashmi Mohan and Shrikant Srinivas Naidu from Yahoo Research and Development, India for providing their valuable suggestions and comments.

6. REFERENCES

[1] Computing Community Consortium (CCC). Challenges and Opportunities with Big Data. [White paper]. February 2012.

[2] Andrew B. King. Types of Web Analytics Software. In Website Optimization. July 2008. O'Reilly Media.

[3] Oracle. Big Data Analytics - Advanced Analytics in Oracle Database. [White paper]. March 2013.

[4] Nikolay Laptev, Kai Zeng and Carlo Zaniolo. Very fast estimation for result and accuracy of big data analytics: The EARL system. In ICDE 2013, pages 1296−1299 .