Big Data Analysis Combining Website Visit Logs with User Segments and Website Content

Matic Kladnik
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
matic.kladnik@ijs.si

Blaž Fortuna
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
blaz.fortuna@ijs.si

Pat Moore
Bloomberg LP
New York, USA
pmoore26@bloomberg.net

ABSTRACT
This paper provides three use-cases of combining website visit logs with user segment data, website content and stock volume data. The use-cases provide concrete examples showcasing how to derive insights into behavior of users on the website. We also present how to efficiently derive required data using MapReduce technique, and implement the use-cases using QMiner data analytics platform.

Categories and Subject Descriptors
Information Systems → World Wide Web → Web Mining

General Terms
Algorithms, Management

Keywords
Web log analysis, Big data, MapReduce, Hadoop, Hive, QMiner

1. INTRODUCTION
Visitors on modern websites leave behind large amounts of traces in form of a web server log, query logs, pixel tracking logs, etc. Website owners analyze these traces to better understand how visitors navigate their website, identify their interests, and optimize advertising opportunities. Most typical use-cases are covered by end-user tools such as Google Analytics, ComScore, Omniture, etc. More sophisticated use-cases, which require custom or ad-hoc processing of the data, are covered by big data frameworks such as MapReduce and its implementations such as Apache Hadoop. However, standard tools provide little or no support for analysis that requires combining visit logs with content presented on the website. In this paper we present three use-cases which combine big data frameworks and text mining approaches with the goal of deeper understanding of visitors and their habits.

Paper is organized as follows. First we provide a quick overview of the input data, and outline how we prepare it for the use-cases using MapReduce framework. We conclude by outlining three use-cases applied to data from a large news website: comparing user segments, identifying user segments by example article and correlating company news visit logs with stock volume.

2. INPUT DATA

2.1 Website Visit Logs
Visit logs provide log of page views on a specific website. Each page view is described by the time, user ID (stored as a first-party cookie), page URL, referral URL, user agent, etc. Such information is typically collected using a pixel tracking mechanism and is commonly used by all standard web analytics tools. In use-cases we use ComScore as the data provider,

2.2 User Segment Data
Web ecosystem allows for deeper annotation of users based on their behavior around the web. For example, visit to a car dealership website can be noted by a third-party data provider, which has an agreement and some tracking mechanism installed on the dealership's website. Such *user segment* data is further aggregated and distributed by data providers such as Krux. Connecting visitors to our website with external database of user segments can provide a much richer understanding of our audience and, as we will see in the first use-case, allows for some interesting analysis. Example of user segments covered include estimated household income, gender, estimated home net worth and others.

2.3 Website Content
Content of the pages on our website provide additional source of data we can use in the analysis when joined with visit logs and user segment data. Content can be represented on different levels of granularity: words, entities, topics, etc.

In our use-cases we collect the content by crawling the URLs mentioned in the visit logs and extracting their content by removing boilerplate. Each page is processed using a standard NLP pipeline [1], providing us with list of topics, named entities and tickers (stocks from companies).

2.4 Stock Volume Data
Our last use-case combines visit logs with market data. We use price and trading volume data provided by Kibot. In this paper we will be using data in hourly intervals, meaning that trade volume values are accumulated by hours.

3. PROCESSING
We process input data using MapReduce parallelization paradigm. MapReduce processes the parts of data separately in the *map* phase and later joins the partial results in the *reduce* phase. This allows us to easily split processing into multiple computing units that can be executed in parallel. In this paper we use Apache Hadoop [2] as MapReduce implementation and Apache Hive [3], which allows us to execute SQL-like queries on Hadoop.

3.1 Aggregating Visit Logs and Segment Data
First we address the task of joining visit logs with user segment data using a cookie shared by both datasets. Join can be executed using the query presented in Figure 1. The query joins three tables: a table with the visit logs, a table linking user IDs from visit logs and segment IDs from segment data, and a table providing segment description.

We tested the query on one week of visit logs on several cluster compositions and the results are presented in Table 1. We can see how the performance of the cluster improves when adding additional instances. However, when going from 4 to 6 instances, we can see a smaller deduction in time taken. That is due to the last

reduce process taking similar amount of time with any number of instances.

Table 1. Performance analysis on segment query

Instances	Duration	Improvement
2	3104 seconds	--
4	1763 seconds	x1.76
6	1427 seconds	x2.18

```
INSERT INTO TABLE visitors_segids
SELECT seg_userid, segid
FROM visit_logs t1, segment_data t2
WHERE t1.seg_userid = t2.seg_userid;

INSERT INTO TABLE visitors_segvals
SELECT seg_userid, segid, segment_title
FROM visitors_segids t1, sikdd_segments t2
WHERE t1.segid = t2.segid;
```

Figure 1. The query creates mapping between user IDs and segment names by joining visit logs and data segments.

```
INSERT INTO TABLE user
SELECT userid, collect_set(url)
FROM sikdd_visit_logs
GROUP BY userid;
```

Figure 2. Aggregating visited URLs by users IDs.

3.2 Aggregating Visit Logs and Content

We now address the task of aggregating visited pages by user. We achieve this by simply grouping visit logs around user ID as show in Figure 2 with the following example query.

Results of the query are inserted into a previously created table. We are calling the *collect_set* function, which returns an array of targeted column values when grouping the results. This way we get a column with distinct user ID values and all connected URLs. In Table 2 we compare performance on this query when for several cluster compositions and can see linear improvement as we increase the number of instances.

In Table 3 we compare this with the performance of the cluster on internal Hive table, as opposed to ad-hoc table pulled from some external source (e.g. Amazon S3). As we can see, the performance can be boosted by reading data from an internal table. When using an internal table, the data is stored locally on the instance. External tables are useful for reading data from an external source (e.g. AWS S3 service), which has to support the HDFS (Hadoop File System). External tables can still be stored on the local HDFS of the instance. Copying data from an external source to an internal table comes in handy when a certain table is used frequently, otherwise the difference in time taken is overturned by the amount of time needed to copy data from an external data source to an internal table. There is also a question of resources availability as moving data to the local instance can fill a lot of the cluster's available storage space. In some cases, the data is simply too large to be moved to the cluster's storage.

As we can see in Table 1, Table 2 and Table 3, adding additional task instances to the cluster can greatly affect the performance. However, at some point performance is improved less effectively when adding additional task instances. This depends on the amount of data, number of files the data is stored in and types of task instances. The last reduce task usually takes some time to complete and is processed on a single instance, which is why it's time taken to process cannot be improved.

Table 2. Performance on visit logs data in external table

Instances	Duration	Improvement
2	713 seconds	--
4	446 seconds	x1.60
6	344 seconds	X2.07

Table 3. Performance on visit logs data in internal table

Instances	Time taken	Improvement
2	275 seconds	--
4	265 seconds	x1.04
6	234 seconds	x1.18

4. USE CASES

In this section we present three use-cases that combine visit logs, segment data, content and stock volume data. Use-cases were implemented in QMiner [4], a data analytics platform for processing large-scale real-time streams containing structured and unstructured data.

4.1 Comparing User Segments

The goal of this task is to compare behavior of different user segments on the website. Segments are provided by User Segment data, which we joined with visit logs and website content. The use-case is prototyped as a web app using QMiner and Node.JS.

Figure 3 shows the interface where the user can select which segments of users should be compared. Query is specified as a collection of segments defining a subset of website visitors. The second group can be either a complement of the first group, or can also be defined through another list of segments.

In the example we compare users identified as engineers versus users identified as administrators. We can see results for this query in Figure 4. Report shows some overall statistics and the odds rations that are significant for the first group. The output can be directly transformed into instructions for an ad server.

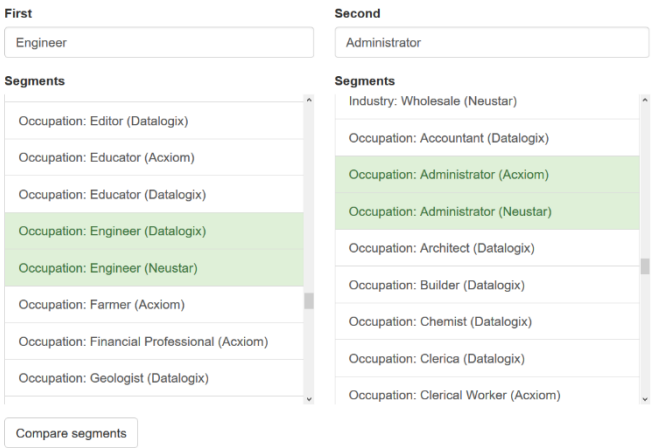

Figure 3. Interface for selecting which segments we want to compare. E.g. engineer vs. administrator

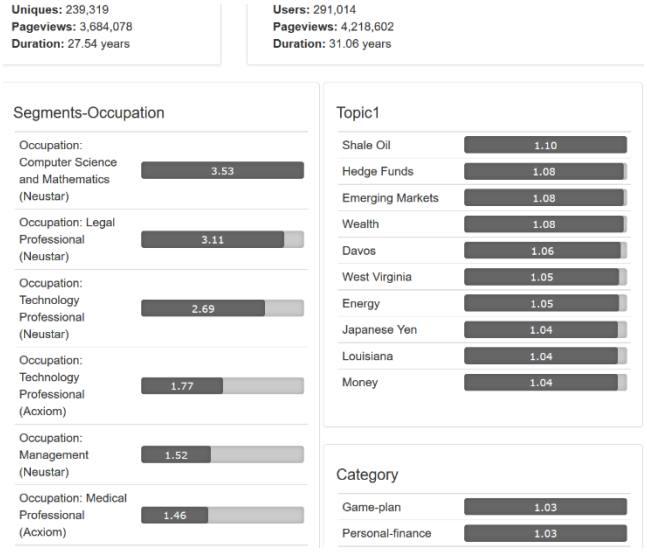

Figure 4. Segment query results

4.2 Analyzing Interests for Topics

The goal of this task is to understand who is interested in specific topics. Topic can be either selected from a predefined list of categories (e.g. Advertisement), or defined via a document. For example, we can search for articles on Product Innovation, which is not included in the predefined list, by using Wikipedia page on product innovation as a query.

Result of such as query is a subset of pages from the website that fit the given topic. We can then join this by using mapping from Section 3.2 to obtain list of visitors that read this content, and, by using mapping from Section 3.1, also their segments. Note that these joins can be executed on one month of data in QMiner in real-near time.

Result is several reports. First, we can generate same kind of report as shown in Figure 4, by contrasting readers of identified pages with the rest of the website's visitors. Second, we can aggregate visited pages and their content in order to identify top topics, people and keywords, as presented in Figure 5 and Figure 6.

4.3 Comparing traffic and stock volume data

As the last use-case we will check for correlation between the stock volume data and a combination of visits logs and content pages. All examples will be focused on AAPL (Apple Inc.) and we will look at the data aggregated by hour.

We start by creating two time-series from the visit logs: (a) number of visits to AAPL quote page, and (b) number of visits to articles related to Apple. First time-series we can obtain directly from visit logs. For the second, we have to first identify significant mentions of entity Apple in the news articles and check their visit statistics. We compere these with a trade volume of AAPL stock. The intuition being, that more news there is about a specific company, the more trading there will be with their stock.

First we compare quote page visits with the trading volume. As we can see in Figure 8, stock volume values fluctuate in similar patterns and time intervals as the number of requests values. Most of the trends can be explained by daily patterns, where most of trading happens during US trading hours, when also website traffic spikes.

If we compare article visits with the trading volume, we can see a clear discrepancy with articles generating more traffic over first few days and the stock having higher trade volume over last few days.

Figure 5. Results for querying on Advertisement input text

Figure 6. Word cloud with top keywords from pages discussing the topic of Advertisement.

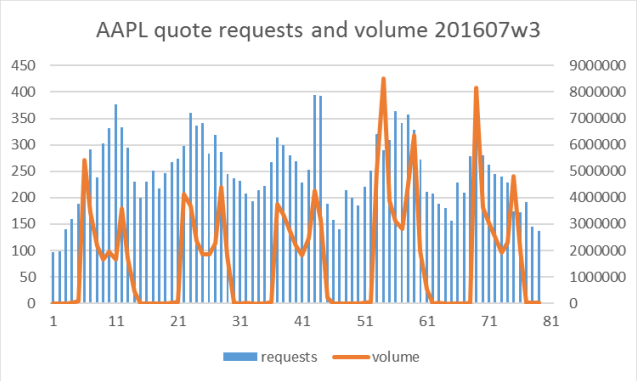

Figure 7. AAPL quote requests and volume

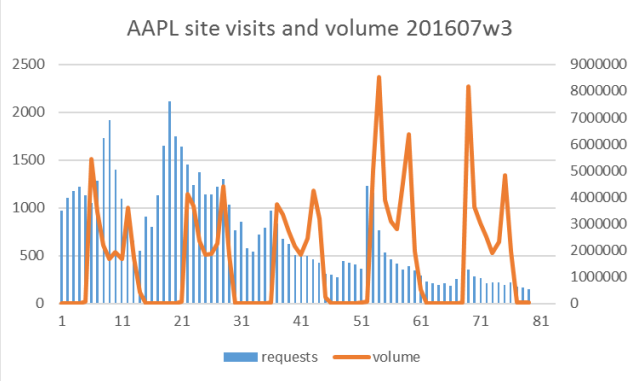

Figure 8. AAPL site visits and volume

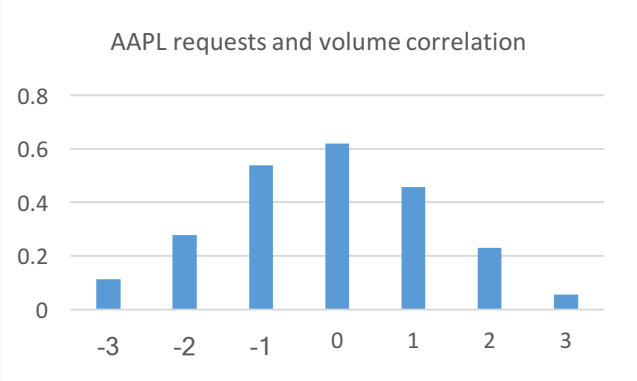

Figure 9. AAPL requests and stock volume correlation

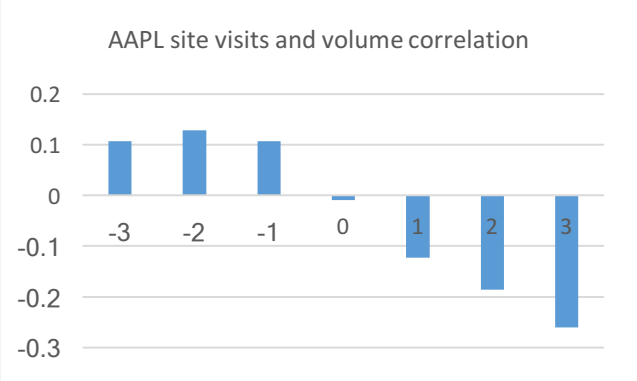

Figure 10. AAPL site visits and stock volume correlation

We can also check the correlation between the trade volumes and the visit data. Figure 9 shows us correlation between quote requests and stock volume. The x-axis represents the offset of request time based on the volume values time, measured in hours. For example, when the offset is -1, we compares quote request data at (h – 1) hour with stock volume data at h hour. The biggest correlation between number of quote requests and stock volume values is within the same hour (no offset). In this case, correlation coefficient is significant as the precise value is 0.62. Whereas if we take the number of quote requests 1 hour in advance of the volume values, we still get a significant correlation coefficient of 0.46, or even greater (0.54) if we take quote requests from 1-hour prior of the volume values. The correlation coefficients fall more, the higher the offset we take between stock volume data values and quote request values. In any case we can observe that correlation between both data is significantly high.

Correlation coefficients on Figure 10 tell us that there is some slight correlation between the numbere of visits to sites that mention Apple Inc. (AAPL) and the volume of AAPL stock on stock exchanges when taking visits prior to the volume movement into account. Similar to Figure 9, the x-axis on Figure 10 represents offset time of site visits, based on the time in stock volume data. There is a small jump in correlation, when taking site visit values that are offset by 2 hours into the past, compared to the stock volume data time. Altough, even at that point the correlation coefficient is 0.13. There is a negative correlation when taking site visit values from the hours following the time of stock market volume values. We can conclude that the correlation between site qoute requests and stock volume data is much stronger than correlation between site visits and stock volume data. However, we can also make an observation that we only get a positive correlation between site visits and stock volume when taking the numbers of site visits that occurred prior to stock volume into account.

5. REFERENCES

[1] Štajner, Tadej, et. al. A service oriented framework for natural language text enrichment. Informatica (Ljublj.), 2010, vol. 34, no. 3, 307-313

[2] Apache Hadoop: http://hadoop.apache.org

[3] Apache Hive: https://hive.apache.org

[4] Fortuna, Blaz, et al. QMiner – Data Analytics Platform for Processing Streams of Structured and Unstructured Data. Software Engineering for Machine Learning Workshop, Neural Information Processing Systems 2014