Senior Thesis Winter Summary

TODO: ACM Format - Double column

ACM In-text citation to be fixed

Blocks of text to fix

Introduction:

The purpose of the thesis is to find areas of congestion using traffic flow data and determine an appropriate advertisement to be placed at a given location, and if possible, at a given time of day. By using the Hadoop Ecosystem, a common tool for big data analysis, we optimize the placement of advertisements for a given area based on traffic flow. Datasets that are being taken advantage of contain traffic flow data, demographics for a given area, and advertisement data for a city. By finding high congestion times for a certain area and combining the advertisement data relative to the area, the system produces optimal ranges to place advertisements, along with the type of advertisement to place. Possible future work in this area includes evaluating the effectiveness of the placed advertisements by using machine learning.

Motivation:

Many cities and states collect traffic statistics for roads, either to show traffic flow on display boards or to gather statistical information. **Utilizing public domain data to aggregate the data from the datasets is essential for users who are interested in finding trends and patterns that are evident in the dataset or related to each other.** By completing this thesis successfully, the result can be an example for other cities to attempt to record traffic data to find patterns and optimize advertisement locations.

Since the collection and storage of data has become easier with the wireless sensors and the availability of cheaper storage, collecting traffic data within a well-funded location does not pose a problem. The only limitation is the sensors used to collect traffic data. Once a city has the data, providing data to the algorithm should produce a result. **With the creation of the algorithm, inputting traffic data should not be a hassle, and provide useful marketing data for a city.**

Literature Review: (Review ACM format)

Currently, there appears to be no academic research into using big data and coordinating traffic flow with advertisements. However, there has been plenty of academic research and articles involving big data and traffic datasets, along with many articles about the applications of using big data and advertisements.

Smith and Demetsky (1997) discussed the importance of “intelligent transportation systems,” at a time before Google Maps and traffic flows were available on the internet. Their academic paper discusses the need to have forecasting models, using several time prediction techniques, such as machine learning and historical average, to determine traffic forecasting. With the datasets being collected, this project is using historical averaging with the available data. Daas, Puts, Buelens, and van den Hurk (2013) highlight that at the time that big data was heavily viewed from an “IT-perspective” and “focus[ed] on soft- and hardware issues.” In the Big Data case study done by Chen, Pao, Lee (2014) about the study of traffic loop detection data, a successful plot of peak hours and vehicle flow was shown. The case study shows the high potential of finding a trend in the data.

The big data and advertisement side of academic articles talked about the collection and use of user data. Couldry and Turow (2014) elaborate on personalized advertising constantly mining personalized data. The article also looks “more broadly at the consequences of embedding big data use in advertising," which is not of concern for this project. Bughin, Chui, and Manyika (2010) discuss the opportunities companies take with using the data available for a web-based company. The availability of data for marketing is not scarce due to the expanding amount of data, which is to be useful in determining advertisements to display is a certain area. Unfortunately, advertisement data is often well protected and requires subscription to view, which poses a problem for an academic thesis.

Do a follow up Literature Review over Break/Final's week

Hadoop Ecosystem

Apache Hadoop is an "open-source software for reliable, scalable, distributed computing" (H2). The purpose of Hadoop is to have a software library that allows for large data sets to be distributed across a cluster of machines and be processed by simple programming models (H2). Hadoop uses a distributed file system, known as Hadoop Distributed File System (HDFS), to help store "very large files with streaming data access patterns" on a network system (H2).

Tools in the Hadoop Ecosystem

Apache Hadoop has many tools to help analyze and process data. Multiple tools were looked at, to determine the helpfulness in processing and finding patterns in the data. Tools examined were Apache Pig, Apache Hive, and Apache Spark.

Apache Pig is a tool used to help handle processing large datasets. The purpose of Pig was to provide much richer data structures, and allows transformations on the data to be much more powerful. *Pig Latin* is the language used to apply the transformations to the data. *Pig Latin* is a “simple query algebra” that allows transformations of: data, such as merging, filtering, and applying functions to the data records or groupings (P1).

Apache Hive was created to allow users with strong SQL skills to run queries on the huge volumes to data. The original creation for Hive was for processing the “huge volumes of data that Facebook stored in HDFS” (H1). As testing this addition to Hadoop, the tool was not idea for finding an area of congestion of a segment and overlapping with advertisement data. Since the goal was to find the best time of congestion and not to retrieve data easily, this tool would not be ideal.

Apache Spark is large-scale data processing on a cluster computing frameworks. The key feature of Spark is the ability to keep large working data sets in memory between jobs (H1). Spark also includes some build analytics tools, such as: machine learning, graph processing, stream processing, and SQL. The attractive feature of Spark was the machine learning library, which could be used as an aspect of machine learning in the thesis.

Spark’s machine learning library contains algorithms, such as: classification, regression, decision trees, recommendation, clustering, topic modeling, and sequential pattern mining (SP1). The initial attempt of using Spark was extremely difficult, as the language and the tools were beyond the knowledge from independent learning during the thesis. Spark’s machine learning library has available interesting machine learning algorithms, but time limits the complete understanding of the tool. Original applications of the tool were not applicable with the formation of the cleaned data, but as the development of aggregating the traffic data becomes clearer, Spark tools could potentially work back into the data flow.

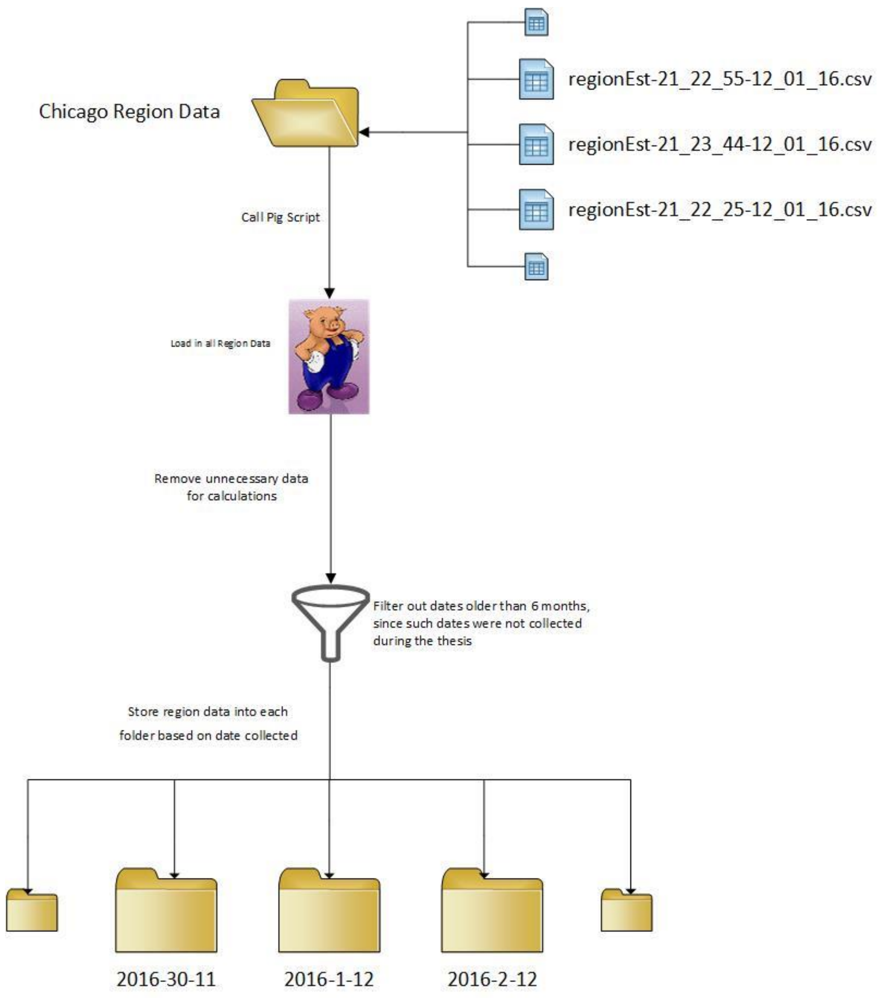
Incorporating Pig

For the thesis, Pig was used to clean and organize the Chicago Traffic region and segment data into an organized data structure, as well as the advertisement data. Region data contained columns not relevant for every time iteration, such as the location and description. Pig combines all the downloaded spreadsheets and removes the unnecessary columns from the Pig data structure, then puts all the regions into a folder based on the date the traffic data stored into the data portal. The segment data was similarly processed, removing unnecessary columns for the congestion calculations, such as location and flow direction, then putting the segments into a folder based on the day of collection.

MapReduce, another tool in Hadoop, has a long process cycle compared to Pig, which involves writing the mappers and reducers, prepares the code for compiling and packaging, submits the jobs, and retrieves the results. To accomplish the same goal as Pig, multiple MapReduce stages would be required to match the multiple mappings and reducing that would fit the data required, but Pig's data flow handles compiling the Pig Latin into MapReduce jobs, which are executed on the Hadoop Cluster. (H1) Therefore, we used Pig as a simpler implementation to clean and process the traffic data.

Figure 1 and Figure 2 both show the data flow of a given Pig script that removes unnecessary columns and data that is older than 6 months. Both flow diagrams have similar structures, since both are accomplishing the same goal. The reason for separate scripts was due to the different column names in each region and segment file, since there is different information needed to define a region or segment.

Figure 1 demonstrates the cleaning and organizing of traffic data for all the region files that have been collected. The region files have 9 columns (Appendix: Chicago Traffic Tracker – Estimates by Region), but only 3 of the columns are important for determining congestion. The region number, speed, and date time are the essential columns needed to find the optimal congestion times for a region. The header column is removed, which is at the top of every file and contains strings labelling the columns. The date and time come in as a string, which is converted into a *datetime* variable when the region number, speed and date time string are pulled from each row. Once the data is ready for output, a folder location is specified and a unique identifier is chosen as a way to store the region rows together based on the timestamp. By extracting the year, day, and month from the *datetime* variable as a string for each row, the strings are concatenated to form the name of a folder in the format of *YEAR-DAY-MONTH*. Cleaning is done to remove any duplicated entries from the condensed data set, as well as any date older than 6 months. Finalizing the script, the multi-sort function uses the concatenated variable date string to store the region, speed, and date time into a folder with other rows falling on the same day.



Chicago Region Data
regionEst-21_22_55-12_01_16.csv
regionEst-21_23_44-12_01_16.csv
regionEst-21_22_25-12_01_16.csv
Call Pig Script
Load in all Region Data
Remove unnecessary data for calculations
Filter out dates older than 6 months, since such dates were not collected during the thesis
Store region data into each folder based on date collected
2016-30-11
2016-1-12
2016-2-12

Figure 1: Data flow of running Pig script on all region data.

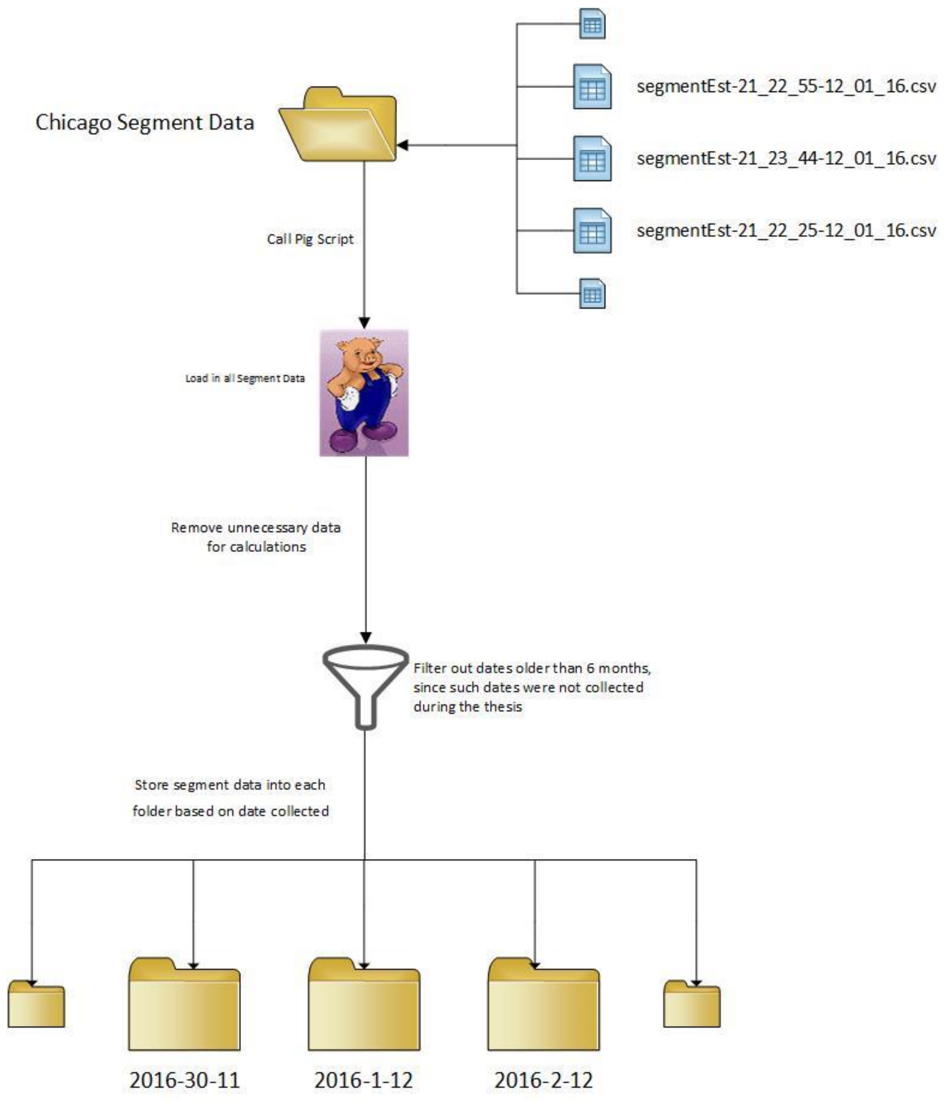

Chicago Segment Data
segmentEst-21_22_55-12_01_16.csv
segmentEst-21_23_44-12_01_16.csv
segmentEst-21_22_25-12_01_16.csv
Call Pig Script
Load in all Segment Data
Remove unnecessary data for calculations
Filter out dates older than 6 months, since such dates were not collected during the thesis
Store segment data into each folder based on date collected
2016-30-11
2016-1-12
2016-2-12

Figure 2: Data flow of running Pig script on all segment data.

Figure 2 demonstrates the cleaning and organizing of traffic data for all the segment files that have been collected. The segment files have 13 columns (Appendix: Chicago Traffic Tracker – Estimates by Segment), but only 3 of the columns are important for determining congestion. The segment id, speed, and date time are the essential columns needed to find the optimal congestion times for a segment. The header column is removed, which is at the top of every file and contains strings labelling the columns. The date and time come in as a string, which is converted into a *datetime* variable when the segment id, speed and date time string are pulled from each row. Once the data is ready for output, a folder location is specified and a unique identifier is chosen as a way to store segments rows based on timestamp. By extracting the year, day, and month from the *datetime* variable as a string for each row, the strings are concatenated to form the name of a folder in the format of *YEAR-DAY-MONTH*. Cleaning is done to remove any duplicated entries from the condensed data set, as well as any rows with dates older than 6 months. Finalizing the script, the multi-sort function uses the concatenated variable date string to store the id, speed, and date time into a folder with other rows falling on the same day.

Current Progress:

The ability to find the best advertisement theme for a given segment is ready. Using a Pig script to clean the data, the advertisement theme data is sorted into a folder based on the segment. Once there, MapReduce jobs determine the best theme by find the averages of themes over a segment and taking the maximum; the advertisement theme to appear next on the billboard is available.

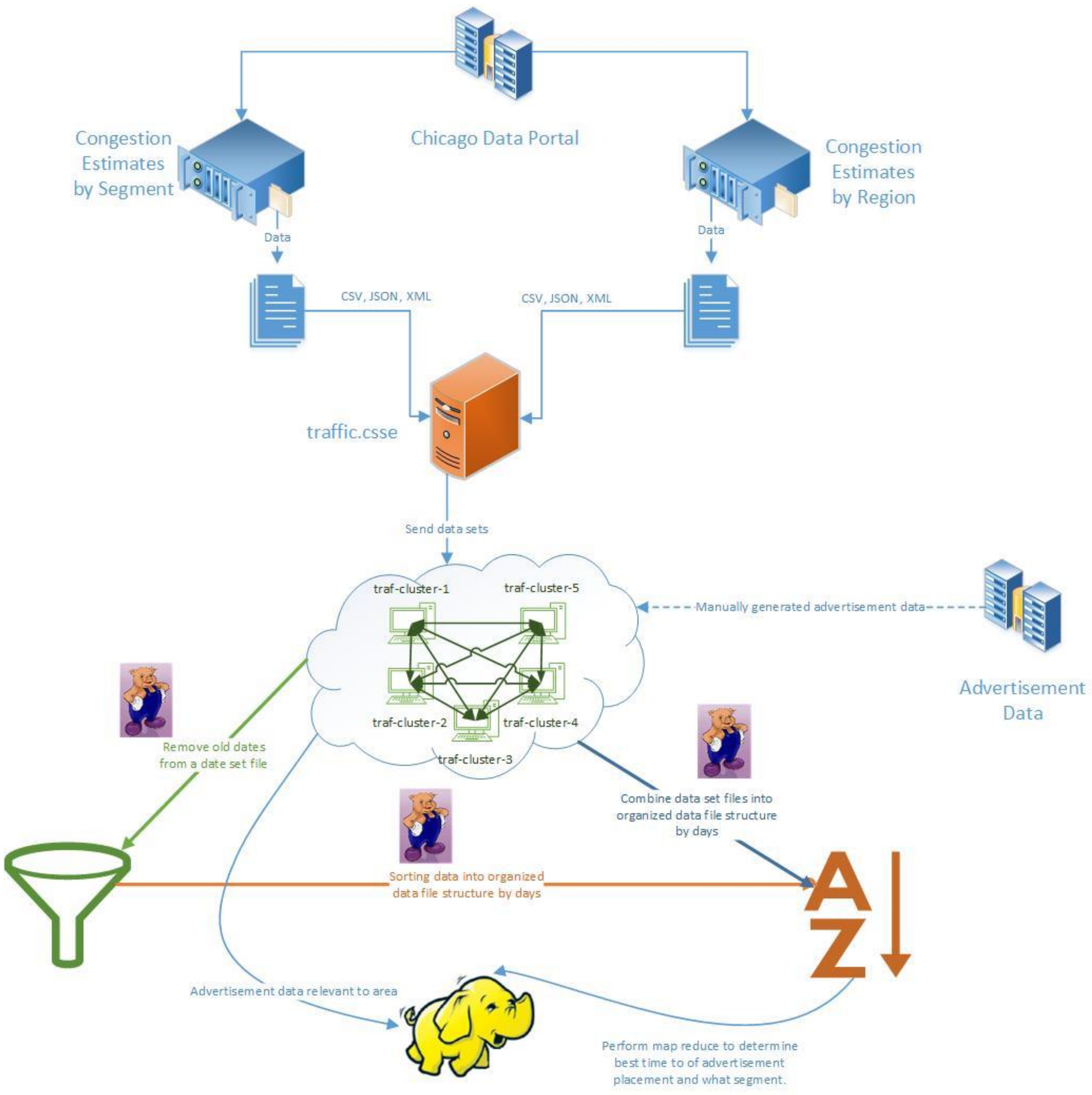
Determining the area of congestion is still a work in progress. The data collection will still be in progress until the end of the quarter. Implementing the MapReduce cycle after the Pig cleaning and processing is the best way of determining to aggregate the data for a given day.

Future Progress:

The remaining goal is to finish implementing the MapReduce job across the segment and region traffic data. Determining the best aggregation to do for the data, along with factoring in the region congestion for the segments in the region will need to be determined. Since a response has not been provided from the Chicago Data Portal dataset manager, determining which segments a region lie in must be found.

The proposed architecture flow is displayed on the next page, showing where the data will be collected, tools used at various stages, along with stages and flow:

Current Architecture:


Chicago Data Portal
Congestion Estimates by Segment
Congestion Estimates by Region
Data
Data
CSV, JSON, XML
CSV, JSON, XML
traffic.csse
Send data sets
traf-cluster-1
traf-cluster-5
traf-cluster-2
traf-cluster-4
traf-cluster-3
Manually generated advertisement data
Advertisement Data
Remove old dates from a date set file
Combine data set files into organized data file structure by days
Sorting data into organized data file structure by days
Advertisement data relevant to area
Perform map reduce to determine best time to of advertisement placement and what segment.

The data flow of the cluster pulls in *Congestion Estimates by Segment* and *Congestion Estimates by Region* from the Chicago Data Portal. Three file formats are downloaded, as back-up copies in alternative forms. The spreadsheet files for regions and segments are transferred to the cluster and stored in HDFS. Once there, the data is cleaned and processed, storing the region, speed, and date time for each row of every region spreadsheet in the appropriate folder. The segment data is cleaned and processed as well, storing the segment id, speed, and date time for each row of every segment spreadsheet in the appropriate folder. The advertisement data is stored in the HDFS, which is processed by a Pig script. The pig script separates the advertisement data by segment, storing advertisement ratings for a given segment into individual folders.

Sources: (Check if current citations were ACM)

Jacques Bughin, Michael Chui, and James Manyika. 2010. Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. *McKinsey Quarterly* (2010), 14.

Anon. Chicago Traffic Tracker - Congestion Estimates by Segments . Retrieved November 1, 2016 from https://data.cityofchicago.org/transportation/chicago-traffic-tracker-congestion-estimates-by-se/n4j6-wkkf/about (S1)

Anon. Chicago Traffic Tracker - Congestion Estimations by Traffic Segments. *Chicago Traffic Tracker - Congestion Estimations by Traffic Segments*. (S2)

Anon. City of Chicago | Data Portal. Retrieved November 7, 2016 from https://data.cityofchicago.org/transportation/chicago-traffic-tracker-congestion-estimates-by-re/t2qc-9pjd/about (S3)

Nick Couldry and Joseph Turow. 2014. Advertising, big data and the clearance of the public realm: marketers' new approaches to the content subsidy. *International Journal of Communication* 8 (2014), 1710–1726.

Piet J.h. Daas, Marco J. Puts, Bart Buelens, and Paul A.m. Van Den Hurk. 2015. Big Data as a Source for Official Statistics. *Journal of Official Statistics* 31, 2 (January 2015). DOI:http://dx.doi.org/10.1515/jos-2015-0016

Brian L. Smith and Michael J. Demetsky. 1997. Traffic Flow Forecasting: Comparison of Modeling Approaches. *Journal of Transportation Engineering* 123, 4 (1997), 261–266. DOI:http://dx.doi.org/10.1061/(asce)0733-947x(1997)123:4(261)

Chen, Xing-Yu, Hsing-Kuo Pao, and Yuh-Jye Lee. "Efficient traffic speed forecasting based on massive heterogenous historical data." *2014 IEEE International Conference on Big Data (Big Data)* (2014): n. pag. Web.

https://cwiki.apache.org/confluence/display/PIG/Index (P1)

http://infolab.stanford.edu/~olston/publications/sigmod08.pdf (P2)

Hadoop: The Definitive Guide – Tom White (H1)

http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F (H2)

https://techcrunch.com/2017/01/08/uber-debuts-movement-a-new-website-offering-access-to-its-traffic-data/#a-bc2ec299-c543-4092-8b33-2f1597b5dd36 (UM1)

https://movement.uber.com/cities (UM2)

http://spark.apache.org/mllib/ (SP1)

Appendix:

Chicago Traffic Tracker – Congestion Estimates by Segments:

Spreadsheet Information:

- SEGMENT_ID: Unique arbitrary number to represent each segment
- STREET: Street name of the traffic segment
- DIRECTION: Traffic flow direction for the segment
- FROM_STREET: Start street for the segment in the direction of traffic flow
- TO_STREET: End street for the segment in the direction of traffic flow
- LENGTH: Length of segment in miles
- STREET_HEADING: Direction of the "STREET" from the origin point
- START_LATITUDE, START_LONGITUDE, END_LATITUDE, END_LONGITUDE: These four points represent the start and end points of the segment in the direction of traffic flow
- CURRENT_SPEED: Real-time estimated speed in miles per hour
- LAST_UPDATED: Date and time of update to spreadsheet

Chicago Traffic Tracker – Congestion Estimates by Regions:

Spreadsheet Information:

- REGION: Name of the region – made up of the names of the community areas within the region
- REGION_ID: Unique arbitrary number to represent each region
- WEST: Lowest longitude values on the regions boundary
- EAST: Highest longitude values on the regions boundary
- NORTH: Highest latitude value on the regions boundary
- SOUTH: Lowest latitude value on the regions boundary
- DESCRIPTION: Describes the streets that demark the region's boundary
- CURRENT_SPEED: Real-time estimated congested level
- LAST_UPDATE: Time stamp for the latest congestion estimation run.

Hadoop Cluster Information

The *traf-cluster* cluster being used consists of 5 nodes, each with 2 processors, 15 GBs of storage, and 16 GBs of RAM available. Across 5 machines, this totals to 10 processors, 75 GBs of hard disk, and 80 GBs of RAM. The total amount of data collected for Region and Segments is approximately 13 GBs, with less than a gigabyte of advertisement data available, the cluster should have enough space to operate.

Data Description:

The two datasets currently collected are Segment Estimations and Regional Estimations, collected from *Chicago Traffic Tracker – Congestion Estimates by Segments* and *Chicago Traffic Tracker – Congestion Estimates by Regions*, respectively. The advertisement data is described in the section below.

Collecting Traffic Data:

The traffic data is available on the Chicago City Data Portal. The data collected is downloaded with a script that checks the data portal server approximately every 30 minutes for *Congestion Estimates by Segments* and *Congestion Estimates by Regions* for changes. The server does not consistently update either data set, which requires constantly checking that repeated data sets are not downloaded; in order to avoid duplicated and unnecessary storage usage.

Segment Estimations:

The segment estimations "contain the current estimated [vehicle] speed for about 1250 segments covering 200 miles of arterial roads" in Chicago (S1). The spreadsheet apparently does not update every segment, which is due to an uploading problem on the data portal's end. The segment estimations are roughly collected every 20 minutes by the Data Portal. The important information from each row in the dataset are the location and area of the traffic flow, speed of traffic, and the timestamp the data was updated. Since the segments are sub-areas of a region, more options for finding optimal advertisement placement will be available, compared to having only 29 areas to find optimal advertisement placement. Refer to the appendix for information about the columns of the dataset.

Region Estimations:

The region estimations “contain the current estimated congestion for the 29 traffic regions” in Chicago (S3). The spreadsheet has updated all 29 regions. The region estimations are supposed to be collected every 10 minutes by the Data Portal. The important information from each row in the dataset are the boundary of a given region, speed of the traffic, and the timestamp the data was updated. Since the 29 regions each have a given number of segments, a given region will provide an overview of the average speed of all the segments. By having segments in each region, a pattern can be found about the given region, which can be used to find the best advertisement. Once a good fit is found for the region, a good fitting advertisement model can be done on a smaller area in the region, such as a segment. Refer to the appendix for information about the columns of the dataset.

Additional Traffic Dataset:

At the beginning of January, Uber release Uber Movement. Uber Movement allows access to “its data around traffic flow in scores where it operates” (UM1). Originally, the data was used to learn more about urban mobility, and help provide feedback to cities to adapt exist infrastructures (UM2). The data is “anonymized and aggregated” to ensure the protection to the users. Access to the datasets is limited to request only, and a request was sent a couple of days after the news article detailing Uber Movement was published. There has been no response about obtaining access.

Advertisement Data:

Available advertisement data was not easy to come by for the Chicago, especially free for academic use. The available data on the Chicago Data Portal did not have a dataset that was easy to identify the locations of the billboards in the city of Chicago. As a substitute, an assumption was made that every segment had a billboard located within the area of the segment, to make it easier to determine if a billboard theme would be successful. To incorporate the advertisement theme of the billboard, data was generated representing the previous advertisement themes that appeared on the billboard along with the success of the theme on the billboard. The themes used were: automotive, clothing, educational, cellular service, technology, medical, job, recruitment, housing, religious, travel, political, theater, restaurant, grocery, insurance, mobile devices, toiletries, lawyer, movie, and television. The use of an electronic billboard, which has rotating advertisements, was not included since there was not more accurate traffic data to localize times for rotating advertisements and duration of an advertisement on the billboard.

The intention of using census data, along with data for a given zip code, was attempted as the source of advertisement data. Due to the large overlap of zip codes and regions, determining which census and economic data to use from all the zip codes and factoring in the amount of the zip code in each region was not a simple task for a beginner using Hadoop. Also, determining the amount of area of a zip code in a region would be difficult to calculate, as some zip codes are not squares (Figure 3).

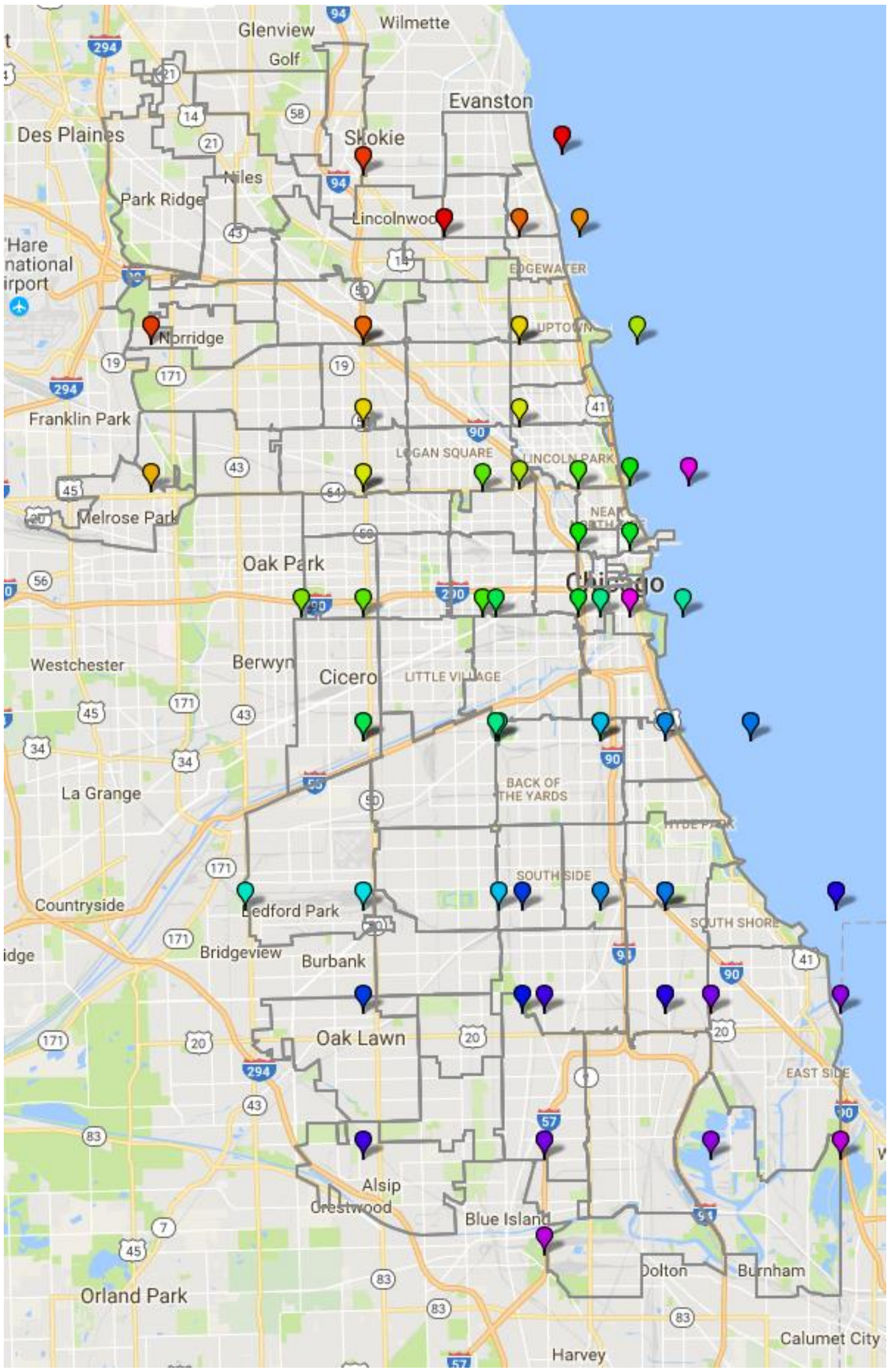


Figure 3: Visual representation of overlap of regions and zip codes. The color dots each represent the bottom left corner and the top right corner of each region. The light gray lines represent the boundary of a zip code.