Traffic Congestion Prediction

Naga Jyothi Kunaparaju

Saint Peter's University nkunaparaju@saintpeters.edu

1-347-341-3307

1 Contribution of Competitor's Article

Smart cities contain huge data stored digitally and numbers of objects connected online. City Pulse project services smart city solution, by means of interpreting big data from Internet of Things and social networks. This project focus on Arhus city sensor data sources. City Pulse Project mainly concentrate on three main areas of Aarhus city road data set, which are Smart Parking, Smart Travel Plan and Smart traffic. Traffic Dashboard solves some of the problems. City Pulse Dashboard is an application to visualize dynamic traffic data. The visualizations are based on current context of the city metrics like Events, Traffic, Parking etc. Historic data used to analyze previous traffic conditions.

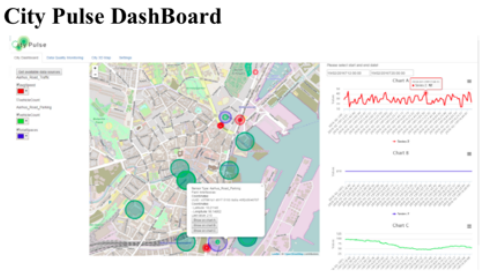

Figure 1 City Pulse dashboard application..

2 Description of Your Contribution

This paper details how technology used in building smart city project efficiently. Similar to City Pulse this project involved in the creation of Smart City application using Aarhus city dataset. But in City pulse project, there is lack of support for prediction strategy based on historic data. Collecting and maintaining historic data is a huge task. Building predictive modeling based on already existing infrastructure give add on advantage to these applications to prevent future traffic problems Similar to City Dashboard project, we focused on city traffic information to predict future traffic conditions based on Time variant and Spatial data. Our assumptions in this project are driving situations varies for different places and also different times. Driving conditions based on routes identified by some important features can be studied to make future predictions. Our system will extend the state of the art City pulse application in two ways. One way by providing prediction capability of future traffic condition and congestion. Other by improving the user experience. Data filtered, formatted and aggregated within Spark framework using Zeppelin, which provides high performance and easy to use built-in infrastructure. In this project, we are going to apply K-means clustering algorithm to create similarity groups to recognize patterns. Another technique we are implementing is ARIMA (autoregressive integrated moving average) time series model to predict future traffic. This analysis is helpful in many areas like travel management systems and safety management.

3 Data Source and Content

Aarhus city traffic data is a collection of datasets of vehicle traffic and street information, observed between two points of a street for a set duration of time over a period of twelve months. There are 449 observation points. This information stored in two data sets. One is metadata, other one is traffic data. Each record in traffic dataset is an observation of how many vehicles passed between two points and its average speed within a five-minute interval. Metadata consists of geographic details such as Latitude, longitude, street name, address, zip code, the city of each road along with exit information. There is a separate metadata file for each city. These two datasets combine with report_ID. Figures 2&3 are snapshot of two datasets. Joining both datasets will give us the ability to aggregate, filter, format data using the spark.

```
%pyspark

df1.show()

+------+---------------+--------+-----+------------------+-------------------+------------+------+---------+
|status|avgMeasuredTime|avgSpeed|extID|medianMeasuredTime|          TIMESTAMP|vehicleCount|   _id|REPORT_ID|
+------+---------------+--------+-----+------------------+-------------------+------------+------+---------+
|    OK|             92|      91|  646|                92|2014-02-13T11:30:00|          35|189978|   158446|
|    OK|             84|     100|  646|                84|2014-02-13T11:35:00|          26|190427|   158446|
|    OK|             78|     108|  646|                78|2014-02-13T11:40:00|          31|190876|   158446|
|    OK|             79|     106|  646|                79|2014-02-13T11:45:00|          28|191325|   158446|
|    OK|             77|     109|  646|                77|2014-02-13T11:50:00|          35|191774|   158446|
|    OK|             83|     101|  646|                83|2014-02-13T11:55:00|          36|192223|   158446|
|    OK|             83|     101|  646|                83|2014-02-13T12:00:00|          24|192672|   158446|
```

Figure 2 Road traffic data

```
%pyspark
df = df1.join(df2, (df1.REPORT_ID == df2.REPORT_ID) ).drop(df1.REPORT_ID)
df.show()

|status|avgMeasuredTime|avgSpeed|extID|medianMeasuredTime|          TIMESTAMP|vehicleCount|   _id|    POINT_1_STREET|DURATION_IN_SEC|POINT_1_NAME|POINT_1_CITY|POINT_2_NAME|       POINT_2_LNG|     POINT_2_STREET|NDT_IN_KMH|POINT_2_POSTAL_CODE|POINT_2_COUNTRY|POINT_1_STREET_NUMBER|ORGANISATION|  POINT_1_LAT|       POINT_2_LAT|POINT_1_POSTAL_CODE|POINT_2_STREET_NUMBER|POINT_2_CITY| ROAD_TYPE|       POINT_1_LNG|REPORT_ID|POINT_1_COUNTRY|DISTANCE_IN_METERS|          REPORT_NAME|RBA_ID|
|    OK|             92|      91|  646|                92|2014-02-13T11:30:00|          35|189978|Østjyske Motorvej|             75|        4321|     Tilst|        4323|10.12519614484404|Nordjyske Motorvej|       113|               8382|        Denmark|                    0|        COWI|56.2173171142913|56.23490211108693|               8381|                    0|    Hinnerup|MAJOR_ROAD|10.10711200000003|   158446|        Denmark|              2345|AAR_BT_0005 4321 ...|159619|
|    OK|             84|     100|  646|                84|2014-02-13T11:35:00|          26|190427|Østjyske Motorvej|             75|        4321|     Tilst|        4323|10.12519614484404|Nordjyske Motorvej|       113|               8382|        Denmark|                    0|        COWI|56.2173171142913|56.23490211108693|               8381|                    0|    Hinnerup|MAJOR_ROAD|10.10711200000003|   158446|        Denmark|              2345|AAR_BT_0005 4321 ...|159619|
|    OK|             78|     108|  646|                78|2014-02-13T11:40:00|          31|190876|Østjyske Motorvej|             75|        4321|
```

Figure 3Merged Data Set

By running analytics on this data, we will be able to find out some patterns trends of traffic in past for different situations like different weather conditions, in major city events, peak times, holiday period. Some problematic time zones and performance bottlenecks were identified using graphs.

4 Your Method:

Data from sensors mobile, social media are some of sources generating data from smart city services. The complete dataset contains 18 million records. Since the volume of data is big, Apache Spark with Python framework will be best option to implement this project. We develop and deploy the application using Zeppelin. Spark is cluster computing framework designed to handle large datasets.

This project utilizes these traffic properties along with time variant and geographical information to find the traffic congestion situation. Three properties identify traffic flows. Traffic speed, average time and vehicle count. Preprocessing of data explained by figure 4. First data loaded into spark as data frame objects. Data types converted from string to float and timestamp. Both data frames contain REPORT_ID which will be used to merge both datasets. By converting into timestamp, we will be able to abstract data by hour, minute, or day of the week for all kinds of temporal data distributions. This data set much cleaner and concise. It contains much smaller foot print compared to original data sets.

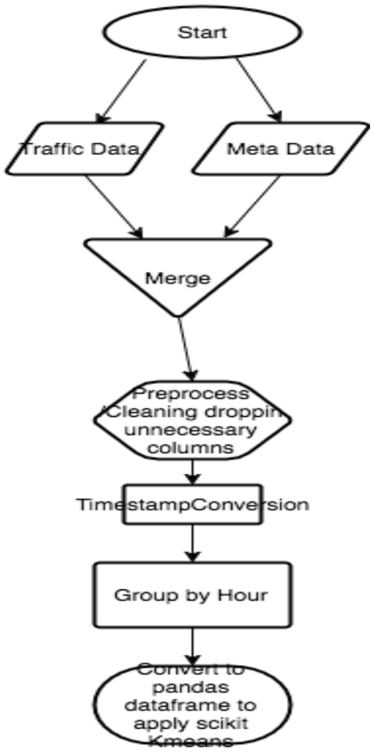

Figure 4 Preprocessing and Data Cleansing

```
%pyspark
rd_data_sel.show()

+--------+----------------+-----------------+----+------+
|avgSpeed|     POINT_1_LAT|      POINT_1_LNG|hour|dayofy|
+--------+----------------+-----------------+----+------+
|      93|56.2139907510439|10.145073305557275|  11|    44|
|      90|56.2139907510439|10.145073305557275|  11|    44|
|      84|56.2139907510439|10.145073305557275|  11|    44|
|      88|56.2139907510439|10.145073305557275|  11|    44|
|      90|56.2139907510439|10.145073305557275|  11|    44|
|      88|56.2139907510439|10.145073305557275|  11|    44|
|      87|56.2139907510439|10.145073305557275|  12|    44|
|      84|56.2139907510439|10.145073305557275|  12|    44|
|      88|56.2139907510439|10.145073305557275|  12|    44|
|      90|56.2139907510439|10.145073305557275|  12|    44|
|      80|56.2139907510439|10.145073305557275|  12|    44|
|      79|56.2139907510439|10.145073305557275|  12|    44|
```

Figure 5 Snap shot of aggregated dataset t

Figure5 shows how the sample data for clustering looks like in Zeppelin. Once preprocessing done we start visualizing data to find patterns.

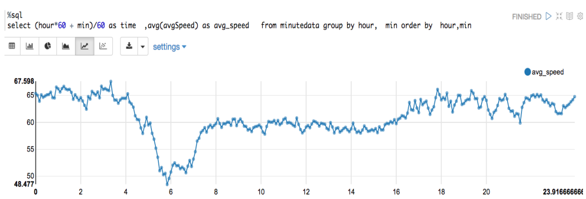

Figure 6 Time series 24 hours average speed

Fig 6 shows time series graph for sample data. We can observe, difference in traffic speed for day and night. This plot gives a glimpse of data that makes us to conclude what model is beneficial for finalizing the model.

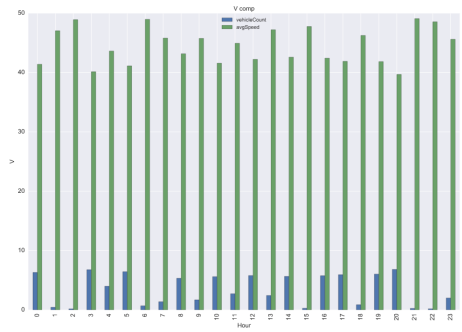

Figure 7 Hourly average speed and vehicle count

Fig 7 Shows there is a strong relationship between number of vehicles in the street and average speed. We can eliminate either one of the variable to build a model. Average speed always more than 40. For that reason, we considered average speed less than 20 might be reasonable estimation of traffic problem. We used this metric in creating Congestion prediction model.

Figure 8 is a heat map. X-axis represents hour of the day. Y-Axis represents Vehicle count. Each cell is for Average speed. By observing we identify clearly, in a day average speed, vehicles travelled in the street are more compared to night.

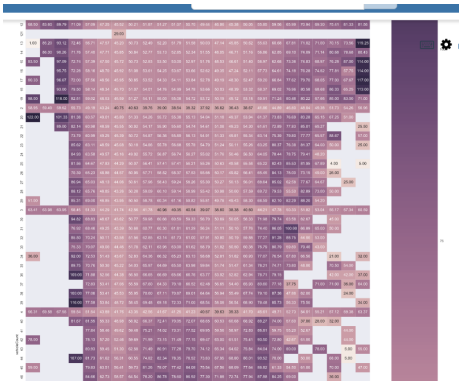

Figure 8 Heat map. Each cell represents average speed. Darker cells with average speed higher than lighter cells. Horizontal axis for number of vehicle's. Verticals time in 24hour format.

K-Means is an efficient method for clustering. Clustering methods are preferred when the outcome is indeterminate but attributes can be divided into separate areas. The second method we are going fallow is Time series analysis. We have traffic data for every 5 minutes. We took one street information ran ARIMA model, and we forecasted average speed.

5 Quantitative Results 1

We will see how to build cluster model and observe results. For this model, the traffic congestion data divided into five time zones (not four clusters), namely, night 12pm - 6am,6am - 10am, 10am - 4 pm, 4 pm - 9pm, 10 pm - 12 pm. Traffic congestion related to certain parameters (such as traffic volume and traffic speed) and also includes many factors. There are two different cases when the vehicle's speed is zero. It may be because traffic blocked by too many vehicles on the road that cannot move or it may also because the street is smooth, with no vehicles driving on the road. By this, a less traffic flow can match two states: the normal flow of heavy traffic or less traffic.

The data is a subset of the complete dataset by average speed less than 20 km/hour and its only 10% of data. Dataset divided into different time zones are ready for clustering.

Elbow plot to obtain best k values is as follows:

```
%r

mydata <- data1
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(mydata,centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```

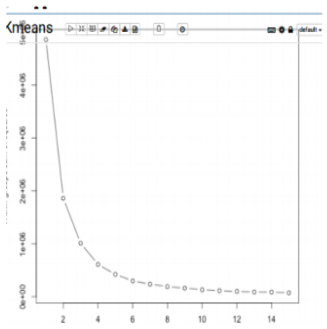

Figure 9 Plot of mean square errors with number of K PLOTTED IN x-axis. The errors drop significantly provide evidence for best K- value.

Figure 10 Time Zone -1 Cluster data representation .Time Zone -1 from night 12 to morning 6AM traffic data.

There is no pattern visible. The less speed areas are may be due to low traffic.

Depending on elbow plot we took k value to be three to classify.

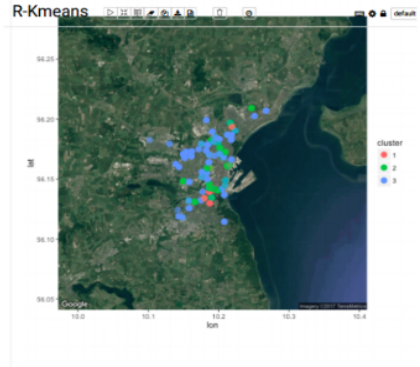

Figure 11 Traffic representation for time Zone -2. Time Zone is from morning 6 Am to 10 Am. More traffic observed in one area.

Figures 10,11,12,13,14, shows geographical maps with plotted clusters. As we can see points in second map are closer than first map. This is because of the fact that before the peak hour, some areas have comparatively higher demand than other areas, mainly the residential communities rather than the major roads.

Figure 12 Time zone-3 from 10 Am to 3PM. Traffic congestion are still high at same place.

At the first time zone, i.e. 00:00~06:00, due to the points are all over the city. But it does not mean that the congestion is severe at that moment. But these may be because of low traffic.

Figure 13 Time Zone -4 Traffic congestion. From 5PM to 10 Pm. Traffic congestion concentration more in one area

Figure 14 Time Zone -5 Traffic congestion. Pattern looks similar to Time Zone-1

In Zones 2, 3, 4 clusters are more concentrated in one area. There a might be a chance that causes for this is more are less by traffic. We have to investigate further whether this road jams because of traffic congestion or not.

6 Quantitative Results 2

We also created a model to estimate future traffic. This we did using Time-series model with R package ASTSA which stands for Applied statistics time series analysis which is very user-friendly in applying the models such as ARIMA. We aggregated data for one road. Data for every five minutes for one year taken for time-series. Feeding this data in different ways and Identifying the right model is kind if Art process.

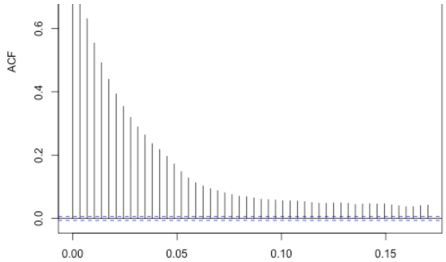

Figure 15Autocorrelation shows it tails of. So MA model is more fit than AR model.

Fig 15 and 16 are autocorrelation and partial autocorrelation graphs. When we closely observe ACF tails off and PACF cutoff. PACF cuts off after 7 lags.

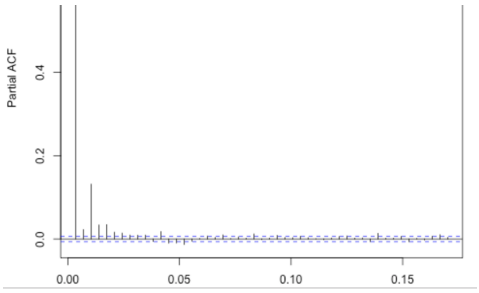

Figure 16 Partial Auto correlation. PACF shoes there are 5 significant lags in the model.

But when we compare these with fig 17 model MA7 is more suitable for data. P values are below confidence interval. A small p-value (for instance, p-value < .05) indicates the possibility of non-zero autocorrelation within the first *m* lags.

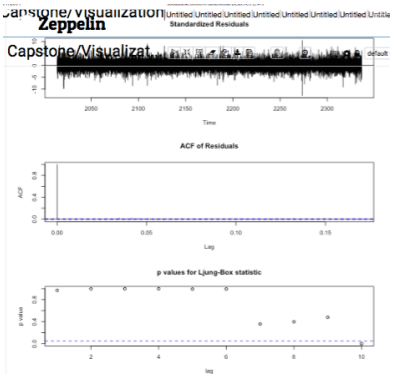

Figure 17 AR7 model we can clearly see the improvement in model in terms of P- values.

Figure 17 clearly displays P value are greater than 0.05 confidence interval show denial of auto correlation, which shows support of the model fitness. Using this model future traffic has been forecasted in figure 20. The prediction very helpful to citizens.

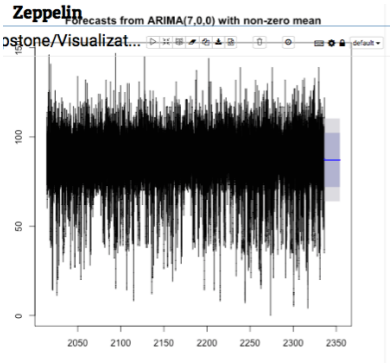

Figure 18 Time series Forecast for next month. Blue area represents forecasted data

7 Discussion: Comparison with Your Competitor

This project aims to provide travel planning solutions, which go beyond the state of the art solutions by allowing users to provide information of traffic congestions. Aim of City Pulse is integrating heterogeneous data streams, run (near-) real-time data analytics. Based on current context of the city routing information is calculated by multidimensional weighting on path, e.g., distance, duration, pollution, events or combined metrics.

Figure 19 Road network by weightage. Rad represents more congestions. Green less and yellow no congestions.

Our metrics for planning to avoidance of routes by historic data. Having insight in to historic data depend on the K means algorithm. K-means similarity measure between two points is obtained by measuring Euclidian distance between routes. It also done by considering all aspects like time for travel, vehicle count, distance geographical and temporal variance. Our frame work is depending on historic data which makes the application decoupled with any other application and able to run separately anytime. For backend we are using Apache Spark, HDFS, Python Pandas, NumPy, SciPy, MLIB, Sci–Kit learn. Our implementation extends the capability and allow users choose daily or hourly aggregation and find the traffic congestions.

8 Performance on Big Data: Time Measurements

S.No	Operation	Time Measured
1	Load raw Traffic data	Load - 41 secs(7 million records)
2	Load Meta data	1 sec(500 records)
3	Union Traffic Data Merge Meta Data	Union – 31sec Merge- data 16 sec
4	Count Data	1 min 35 sec(18 Million records)
5	Aggregate by time	2 sec
6	Filtering and grouping data	2 sec
7	Save formatted data	46sec (2 Million)
8	load formatted data	6min
9	K means -15 times Plot	6 secs(2 Million Records) 10 Secs (Plot + Map)
10	Time series Plot	5secs(1 Million Records) 19secs
11	Plot the results	Bar chart -1 min 5 sec. Heat Map -1min 20 sec

Figure 20 Performance bottlenecks observed during development

Figure 22 summaries crucial operational performances in this project. Using spark automatically parallelizing gives added advantage to process such huge amount of data. Comparing count step(stap-4) with aggregation steps(steps 5&6), count takes more time than other aggregation operations because count is RDD Action and other aggregation operations are RDD transformations. Transformations are much faster as there is no collection on data. Spark only create lineage graph. This clearly indicates advantage of using spark compared to single node computing.

9 Conclusion

In this project we were able to create sophisticated tools like City Pulse Dashboard, which will make citizen life more comfortable and traffic management can take better decisions. We have created a tool set with advanced techniques, using Machine learning algorithms like K-means and ARIMA time-series model. Results came out positive which makes us comfortable in taking a further step in this route. One important issue which will provide more accurate estimation of traffic is how to identify the bottleneck of the urban street network. Finding seasonal patterns also makes better conclusions about traffic. Fine tuning the algorithm with these bottleneck issues will result in more accurate findings. When all these ideas, correctly used with advanced tools, will provide much bigger opportunity to improvement in modeling techniques in future. This, in turn, provide better quality citizen's life by estimating traffic problems in advance.