Highway Tollgates Traffic Flow Prediction

Nikhil Kumar Chandra, Shruti Harihar

1. Introduction

This is a competition held by KDD for the year 2017. According to the competition description and background study highway tollgates are well known bottlenecks in traffic networks. During peak time, huge queues at the tollgates can get little overwhelming for the authorities managing traffic. There is a demand for effective preemptive countermeasures to tackle this problem. The countermeasures can be expediting the toll collection process and streamlining future traffic flow. The expedition of toll collection could be simply allocating temporary toll collectors to open more lanes. These countermeasures work only if we receive a reliable prediction for future traffic flow. For example, if we could predict, heavy traffic for the next hour or two which will enable traffic management authorities to deploy additional toll collectors. In Future, traffic flow could be streamlined by adaptively tweaking traffic signals at upstream intersection.

Traffic flow patterns depends on many factors such as weather, time of the day etc. Predicting traffic flow patterns can help in other advancements in mobility such as sharing accurate results with self-driving cars. Hence machine learning becomes an appropriate tool to solve such problems and apps such as Waze and Google Maps generate huge amount of traffic related data which help us to train machines better.

2. Related Work

A lot of research has already been done in the area of traffic prediction. Eric Horvitz (Technical Fellow, Managing Director at Microsoft Research) and his team researched and worked on traffic congestion problems and they tried to infer and forecast the flows of traffic. The work leverages machine learning to build services that make use of both live streams of information coming from various sensors and large amounts of heterogeneous historical data. This has led to development of multiple prototypes and real-world services such as traffic-sensitive directions in Bing Maps.

Stephen Clark(Research Fellow, Institute for Transport Studies, University of Leeds, UK) provides a solution by applying pattern recognition and pattern matching techniques. It uses a multivariate extension f non-parametric regression that exploits the 3D nature of the traffic state.

The major drawback is that the dataset doesn't provide full topology information about the traffic network and it is relatively new. Hence the techniques might not work as desired. Hence, we try to approach the problem using regression.

3. Dataset and Preprocessing

"Highway Tollgates Traffic Flow Prediction: Travel Time & Traffic Volume" project is a part of the KDD cup competition.

Five datasets are offered in this competition. They are road network topology in the target area, vehicle trajectories information, historical traffic volume at tollgates, and weather data. The detailed information is shown in Table 1.

Link						
No of Attributes			7			
No of Instance			24			
link_id	length	width	lanes	in_top	out_top	lane_width

Route		
No. of Attributes		3
No. of Instance		6
intersection_id	tollgate_id	link_seq

Trajectories					
No. of Attributes			6		
No. of Instance			109244		
intersection_id	tollgate_id	vehicle_id	starting_time	travel_seq	travel_time

History Traffic Volume at Tollgates					
No. of Attributes			6		
No. of Instance			543699		
Time	tollgate_id	direction	vehicle_model	has_etc	vehicle_type

Weather Data								
No. of Attributes					8			
No. of Instance					782			
date	Hour	pressure	sea_pressure	wind_direction	wind_speed	temperature	rel_humidity	precipitation

Table 1: Dataset Information

We cannot use the original dataset to train our regression model because the number of instances varies from table to table. We extract the volume and travel time of each tollgate for a 20-minute timeframe. The attributes colored in blue in table 1 are key attributes which are used to connect datasets for vector creation. The attributes colored in green are potential attributes which have a potential contribution to the result. The extracted data tables are shown below.

Field	*Type*	*Description*
intersection_id	string	Intersection ID
tollgate_id	string	Tollgate ID
time_window	string	e.g., [2016-09-18 08:40:00, 2016-09-18 09:00:00)
avg_travel_time	float	Average travel time in seconds

Table 2: Travel Time from Intersections to Tollgates

Field	*Type*	*Description*
tollgate_id	string	Tollgate ID
time_window	string	e.g., [2016-09-18 08:40:00, 2016-09-18 09:00:00)
direction	string	0: entry, 1: exit
volume	int	total volume

Table 3: Traffic Volume at Tollgates

Once we extract the data for the given time frame (20 minutes), we process the data to get the vectors shown below. These vectors are directly fed into the training models to predict the travel time and volume.

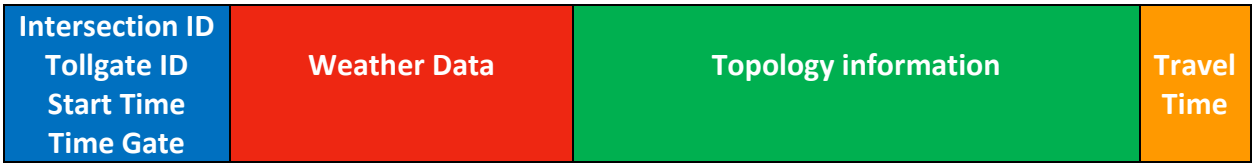

Figure 1: Vector for Travel time

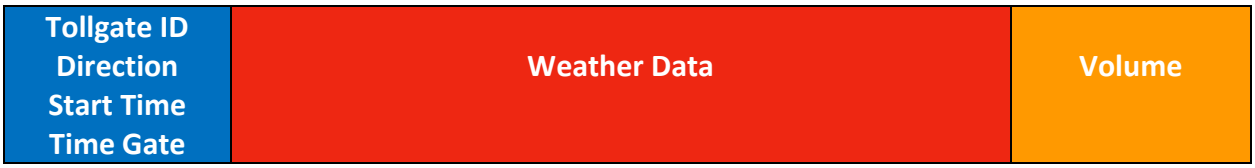

Figure 2: Vector for Volume Prediction

The aggregated vectors of volume and travel time follows the same pattern. Only difference being, we do not consider the topology information in volume task. There are different routes for each tollgate and each route has a different links sequence. This kind of scenarios varies the length of the vectors.

The description of the aggregated vectors is as below.

	Travel Time Vector	*Volume Prediction Vector*
No. of attributes	57	12
No. of instances	25144	10063

Table 4: Description of aggregated vectors

4. Preprocessing techniques

4.1. Normalization:

Normalization is scaling technique, Where, we can find new range from an existing one. We can use it for the prediction or forecasting purpose [3]. As we know there are so many ways to predict or forecast but all can vary with each other a lot. So, to maintain the large variation of prediction and forecasting the Normalization technique is required to make them closer.

$$v_i' = \frac{c_i - \bar{V}_A}{stand\_dev(V_A)}$$

Where
v_i' is Z-score normalized values
v_i us value of the row V_A of ith column

$$stand\_dev(V_A) = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(v_i - \bar{V}_A)^2}$$

$$\bar{V}_A = \frac{1}{n}\sum_{i=1}^{n} v_i$$

In other words, $\bar{V}_A$ indicates the mean of the given attribute and $stand\_dev(V_A)$ indicates the standard deviation of the population. This standard score is widely used in data preprocessing.
It measures the sigma distance of actual data from the average and provides an assessment of how off-target a process is operating.

5. Prediction Model and Methodology

5.1 Decision Tree Regression

Decision tree builds regression or classification models in the form of a tree structure. The dataset is broken down into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. DT's where the target variable can take continuous values are called regression trees. There are variety of decision tree learning algorithm such as ID3, CART and MARS and we can use Information gain(IG), Variance reduction and Gini impurity as metrics to split the tree.

The algorithm used in Decision tree regression and classification tree is one and the same except for the difference that the target value in decision tree regression becomes continuous instead of discrete. Decision models are easy to understand and help us to understand the dataset.

However, one obvious disadvantage of decision tree is overfitting. It sometimes creates complex trees and does not generalize well from the training data. Using various pruning techniques still cannot guarantee model generalization.

5.2 Support Vector Regression (SVR)

Support Vector Machine(SVM) can also be used as a regression method, keeping all the main features that characterize the algorithm (maximizing the margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with some small differences. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

Solution:

$$\arg min \frac{1}{2}||w||^2$$

Constraints:

$$\begin{Bmatrix} y_i - < w, x_i > -b \leq \varepsilon \\ < w, x_i > +b - y_i \leq \varepsilon \end{Bmatrix}$$

x_i = vector of potential attributes
y_i = corresponding target value
ε = Threshold parameter

5.3 Bayesian Ridge regression

Bayesian regression technique can be used to include regularization parameters in the estimation procedure: the regularization parameter is not set in a hard sense but we can tune it according to the data.

Bayesian Ridge estimates a probabilistic model of the regression problem. The prior for the parameter w is given by a spherical Gaussian:

$$p(w|\lambda) = N(w|0, \lambda^{-1} I_P)$$

The priors over α and λ are chosen to be gamma distributions, the conjugate prior for the precision of the Gaussian.

The resulting model is called *Bayesian Ridge Regression*, and is similar to the classical Ridge. The parameters w, α and λ are estimated jointly during the fit of the model.

5.4 K Nearest neighbors

Neighbors-based regression can be used in situations where the data labels are continuous rather than discrete variables. The label assigned to a query point is computed based on the mean of the labels of its nearest neighbors. K Neighbors Regressor implements learning based on the k nearest neighbors of each query point, where k is an integer value specified by the user.

The simple nearest neighbors' regression uses uniform weights: which is, every point in the local neighborhood contributes uniformly to the classification of a query point. Under some circumstances, it can be advantageous to weight points such that nearby points contribute more to the regression than faraway points.

5.5 Ensemble Method

The methods we tried are Bagging and Random forest.

5.5.1 Bagging methods

Bagging methods form a class of many algorithms which build several instances of a black-box estimator on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. We can use these methods to reduce the variance of the base estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it. In many cases, bagging methods constitute a very simple way to improve with respect to a single model, without making it necessary to adapt the underlying base algorithm. As they provide a way to reduce overfitting, bagging methods work best with strong and complex models

5.5.2 Random Forest

In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually increases slightly (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.

6) Evaluation Metrics

We used mean square error(MSE), mean absolute error(MAE) as our basic evaluation metric and mean absolute percentage error(MAPE) as the key point. MSE and MAE is widely extensively used in regression analysis as they provide critical information about the model. Since the competition demanded to use MAPE metric, we will focus more on MAPE than the other two metrics. Below are the details of the metrics mentioned:

6.1 Mean Absolute Percentage Error(MAPE)

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method for example trend estimation. Accuracy is expressed in percentage. It has different definitions for each task. For travel time prediction task, MAPE is given by:

$$MAPE = \frac{1}{R}\sum_{r=1}^{R}(\left(\frac{1}{T}\right)\sum_{t=1}^{T}|\frac{d_{rt} - p_{rt}}{d_{rt}}|)$$

Where d_{rt} and p_{rt} are the actual and predicted average travel time for route r during time window t.

And the MAPE for volume prediction task is computed as:

$$MAPE = \frac{1}{C}\sum_{c=1}^{C}(\left(\frac{1}{T}\right)\sum_{t=1}^{T}|\frac{f_{ct} - p_{ct}}{f_{ct}}|)$$

Where f_{ct} and p_{ct} are the actual and predicted traffic volume for a specific tollgate-direction pair c during time window t. and C is the number of tollgate-direction pairs.

6.2 Mean Square Error (MSE)

The mean squared error (MSE) or mean squared deviation(MSD) of an estimator, measures the average of the squares of the deviations or errors—which is, the difference between the estimator and what is estimated. The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(X_i - Y_i)^2$$

Where $f(X_i)$ is the prediction of instance X_i, and Y_i is the true value of that instance.

6.3 Mean Absolute Error(MAE)

Mean absolute error (MAE) is a measure of difference between two continuous variables. It is a quantity used to measure how close the predictions are to the eventual outcomes.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(X_i) - Y_i|$$

Where $f(X_i)$ is the prediction of instance X_i, and Y_i is the true value of that instance.

7. RESULT

All our code was written in python using the Jupyter Notebook and we used scikit-learn, pandas and numpy packages in this project. Below we have tabulated (table 5 and 6) the best results we achieved for each of the error metric we used. Figure 3 and 4 graphically shows the comparison between the algorithms we used for this project.

	MAPE	*MSE*	*MAE*
Decision Tree Regressor	0.331464636469	5641.92102634	42.1175446429
SVR	0.201875255101	2099.33590758	26.5254086869
Bayesian Ridge	0.210225140542	1781.26996322	25.3077006077
KNN Regression	0.229760059494	2204.57691387	28.4908805893
Random Forest	0.211492883615	1845.66189727	25.8834434225
Bagging	0.201869683844	2098.43455654	26.5200978813

Table 5: Travel time prediction errors

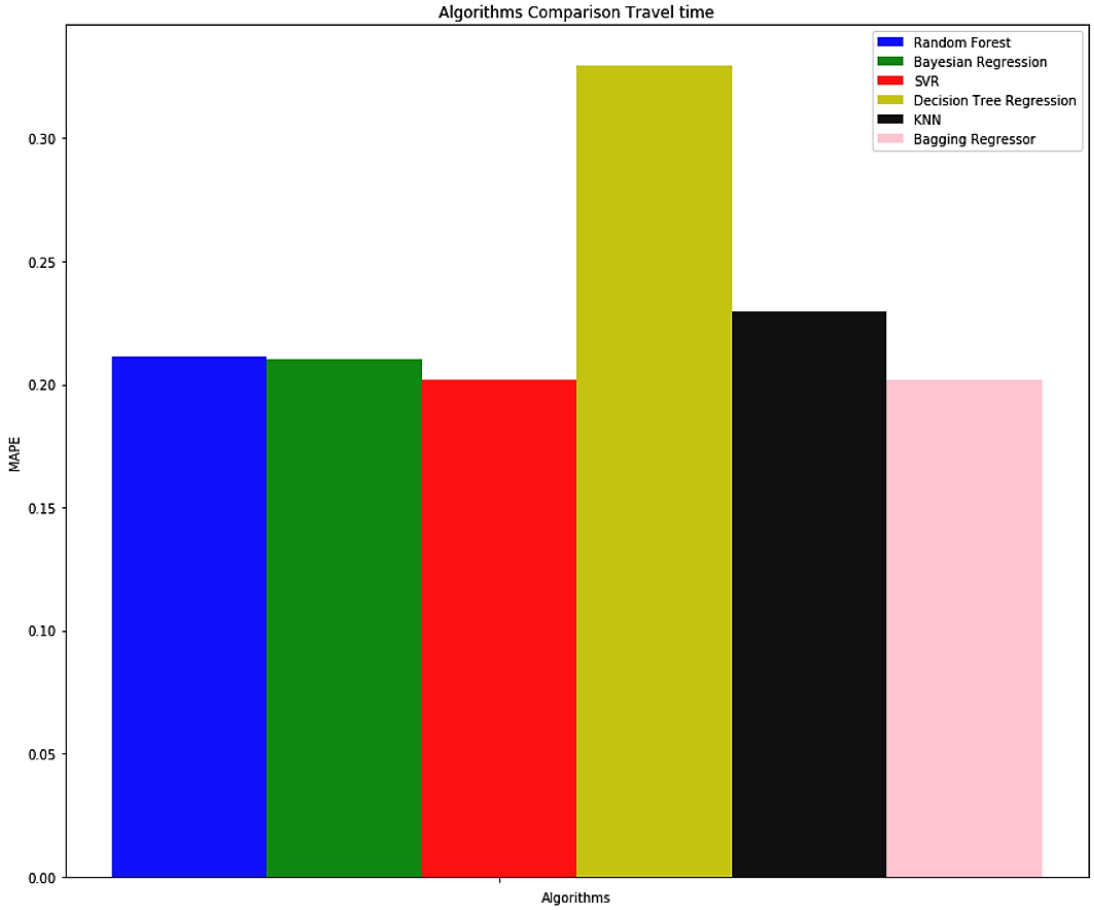

Figure 3: Travel time prediction comparison

	MAPE	*MSE*	*MAE*
Decision Tree Regressor	0.914565112321	3361.97142857	46.8142857143
SVR	0.5586230289	1661.8157026	32.9731943046
Bayesian Ridge	0.615976677619	1465.3170762	31.2057339491
KNN Regression	0.632595341123	1418.96856347	31.4203228571
Random Forest	0.597842390574	1413.57807453	31.1882618011
Bagging	0.557950603152	11661.32985176	32.9588993094

Table 6: Volume prediction Error

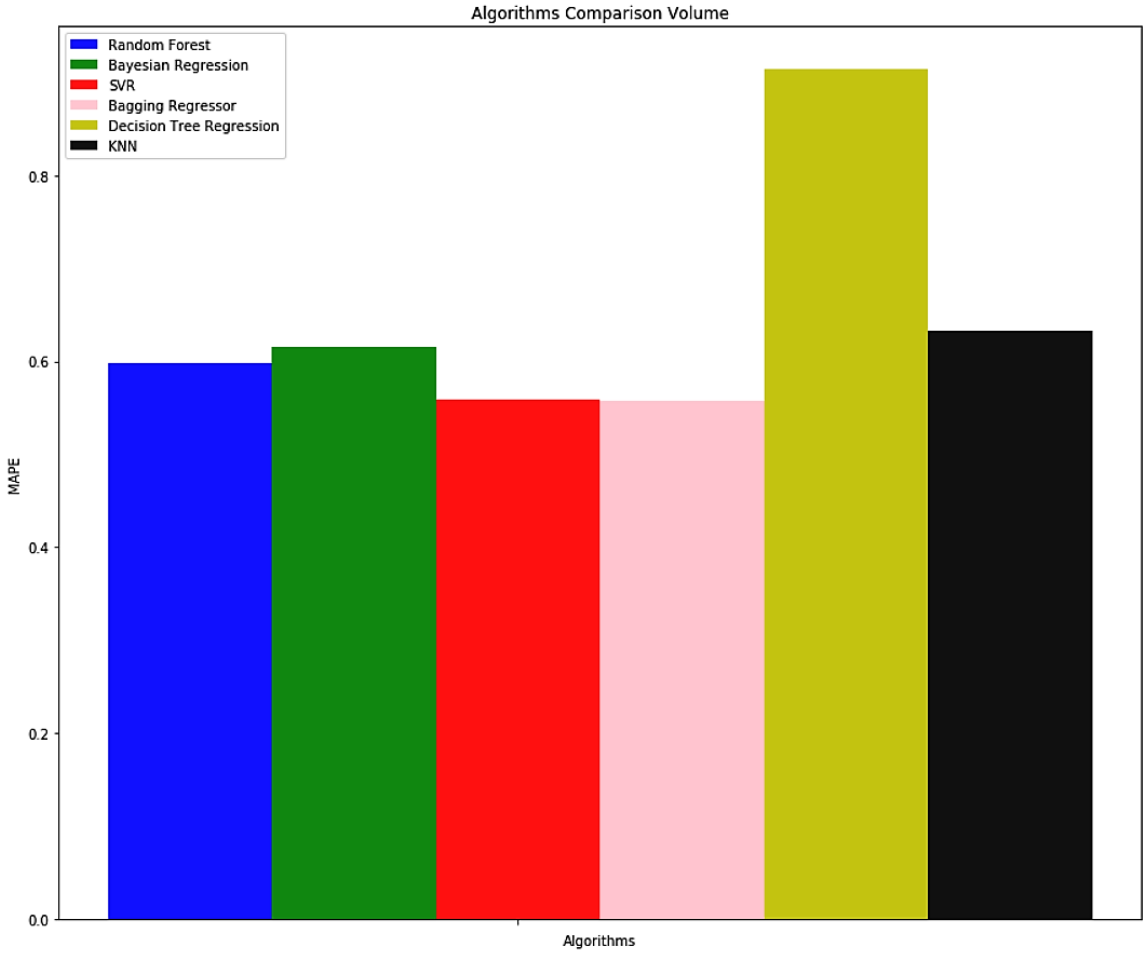

Figure 4: Volume Prediction comparison

It is evident that ensemble technique i.e. Bagging, performed better than the rest of the regression models. We can also observe that SVR works relatively well with both travel and volume vector due to its kernel effect and good approximation with non-linear data. We also observed that Bayesian Ridge regression performed well in the travel time prediction due to its efficiency and robustness with complex data.

8. Conclusion & Future Work

We would like to conclude that we tried many techniques both for preprocessing and regression models. Bagging gave us the best MAPE results. Usually ensemble techniques use better strategies in data sampling which in turn will enhance the result of the regression model.

For the given task, we also can try to divide the vector into different parts and run various regression models on each part of the vector and combine all the results in the end. This technique is appealing since different parts of the vector might influence the target value in different ways. Since we excluded topology information in our volume prediction vector, we can find a way to use that information in the vector and still land up with good results.

9. Reference

[1]https://tianchi.aliyun.com/competition/information.htm?spm=5176.100067.5678.2.8CnCPt&raceId=231597

[2] S.Gopal Krishna Patro, Kishore Kumar sahu. “Normalization: A Preprocessing Stage”. ResearchGate-Publication:274012376

[3] Clark, Stephen. "Traffic prediction using multivariate nonparametric regression." Journal of transportation engineering 129.2 (2003): 161-168

[4] http://scikit-learn.org/stable/supervised_learning.html

[5] https://www.microsoft.com/en-us/research/project/predictive-analytics-for-traffic/

[6] https://www.wikipedia.org/