Predicting freeway traffic in the Bay Area

Jacob Baldwin
Department of Electrical Engineering
Stanford University
Stanford, CA, 94305
Email: jtb5np@stanford.edu

Chen-Hsuan Sun
Department of Electrical Engineering
Stanford University
Stanford, CA, 94305
Email: chsun@stanford.edu

Ya-Ting Wang
Department of Electrical Engineering
Stanford University
Stanford, CA, 94305
Email: yatingw@stanford.edu

***Abstract*—The hourly occupancy rate of a specific location on the I-280 is predicted using both linear regression and functional regression. Linear regression is implemented using successively finer features to improve prediction accuracy. Functional regression is also implemented by using the hourly traffic from one day as a feature to predict the hourly traffic of the next day. A total of four different regressions are proposed, implemented, and compared to observe which ones best predict the aggregate occupancy rate for the given sensor.**

I. Introduction

Prediction of traffic is an area where the application of machine learning could help to pinpoint where additional infrastructure may be the most beneficial. The aim is to build a model of traffic on freeways in the Bay Area, such that we might, based on the time of the day, day of the week, and the weather, identify which sections are congested and estimate delays.

Many different types of models and methods have been developed to model traffic. Traditionally, it has been done with time series analysis using methods such as auto-regressive moving averaging, generalized linear autoregressive moving average models[1] or methods such as those presented in [2]. Machine learning applications involved using Kalman filtering [3] or using artificial neural networks[4]. More advanced methods are described in [5].

Notably, there are also many different settings in which traffic can be modeled where different methods might apply better, for example modeling traffic in an urban setting, or on freeways, or investigating how delays propagate and what causes congestion, or estimate how long it takes for congestion to clear once action has been taken. Some applications of machine learning to this work are described in [6].

It can be seen that traffic modeling is a vast field with many different possible approaches. To limit the difficulty and scope of this project, we choose to simply do some basic modeling of traffic considering only freeways in the Bay area, starting specifically with the I-280, attempting to model how congested they can be using metrics provided in data sets we have found and using this model to predict future traffic levels. Output metric to be considered is aggregate occupancy rate of lanes. Aggregate occupancy rate is a number between 0 and 1 which describes how often that lane is occupied as determined by a sensor at specific time points throughout the day.

II. Data set processing and features

In light of the fact there there are obvious inputs and outputs for which data can be found, we cast the problem as a supervised learning problem and find data for it.

In order to correlate weather data with traffic data, we elected to download data off the California Department of Transportation Performance Measurement System (PEMS) website directly and process it to meet our needs. Weather data was extracted from the National Centers for Environmental Information[8] using their data tool for each day of the year, including precipitation and temperature data.

Data is then downloaded from the PEMS website for a single sensor located on freeway I-280 North in San Francisco. We arbitrarily chose a location in San Francisco, sensor 400156 which corresponds to the section of I-280 passing above Havelock St. PEMS provides both ”aggregate occupancy rate” of lanes on the freeway, as well as ”average speed”. Aggregate occupancy rate is used as an initial starting point for the output we are attempting to model and predict, inspired by the data set provided for a part of the study described in [9] whereby occupancy data was used to predict which day of the week it was.

By combining all of this data, we now have a dataset with the features and output shown in Table I.

TABLE I: Data set format

Feature 1	Feature 2
Day of week	Average Temperature
Feature 3	Output
Precipitation	Average Occupancy Rate

This is the first set of data we attempted to analyse to see if a model can be trained to recognize weekly trends, and if we might better recognize minute differences between those trends with the addition of weather features. Visualisation of the weekly trend is shown in Figure 1. There are some obvious outliers which are the holidays. Therefore, holidays are removed when prediction is performed.

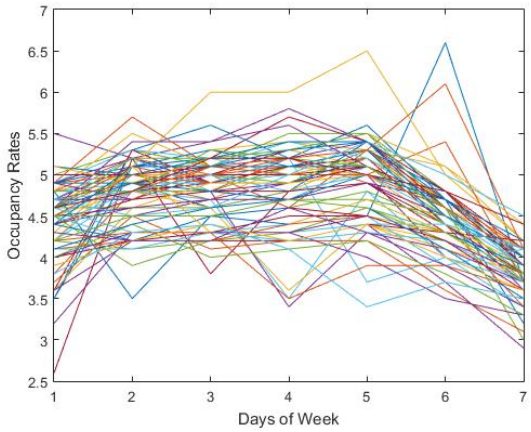

Fig. 1: Visualisation of weekly trend in occupancy rate

The second set of data which are analysed is hourly data instead of daily data mentioned above. Table II shows the features and output we intend to use. An additional feature "Hour" is added to further characterize the hourly trend of a given day of week.

TABLE II: Data set format

Feature 1	Feature 2	Feature 3	Feature 4	Output
Day of week	Hour	Avg. Temp.	Precip.	Avg. Occupancy

The visualisation of this data, which is the hourly trend for a particular day of the week are shown in Figures 2 and 3. This has noticeable more distinct trends, and so the expectation is that a more interesting an useful result can be obtained by looking at this dataset.

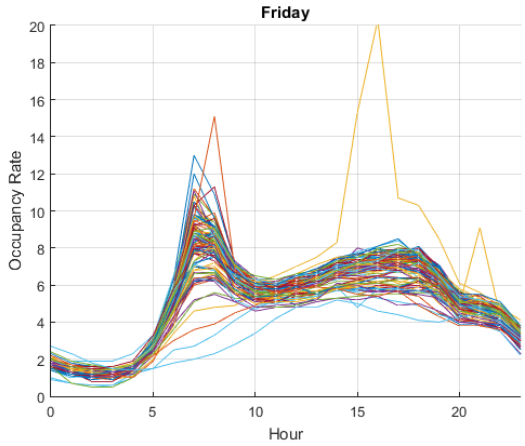

Fig. 2: Visualisation of Friday trend in occupancy rate

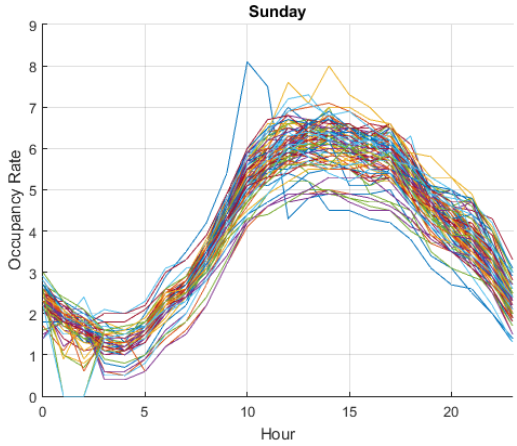

Fig. 3: Visualisation of Sunday trend in occupancy rate

Initial analysis was performed on a data set spanning only two years. In the final stages of this project, we chose to collect and process five years of data for the single sensor chosen above, in order to compare the effectiveness of different methods.

III. Process and General Methodology

An obvious approach is to model this using linear regression, by fitting a trend to the available data. Methods proposed are:

- Linear regression
- Locally weighted linear regression

Both of these were first implemented on the data set with daily aggregate occupancy rate data to see if a useful weekly trend could be recognized, and a decent fit to actual occupancy rates made. For our first attempt, they have been performed on daily average occupancy rate, and so are trying to find a weekly trend for the first dataset described in Table I. The result is illustrated in Figure4, where it can be seen that when used on weekly data, we more or less are just predicting an "average". The introduction of weather data simply adds small perturbations to this average which really doesn't appear to have any correlation with actual occupancy rate.

We conclude that average daily occupancy rate may not be a good metric. Instead of training the linear regression model on the whole week at once, the idea is to train a different model for each day of the week based on the features available. Thus, we takes input a day of the week as a feature, then using that to select the appropriate model for fitting weather data in order to determine the expected occupancy rate. The following sections give detailed analyses and results for using hourly occupancy rate as output. Methods include linear regressions with different features and functional regression.

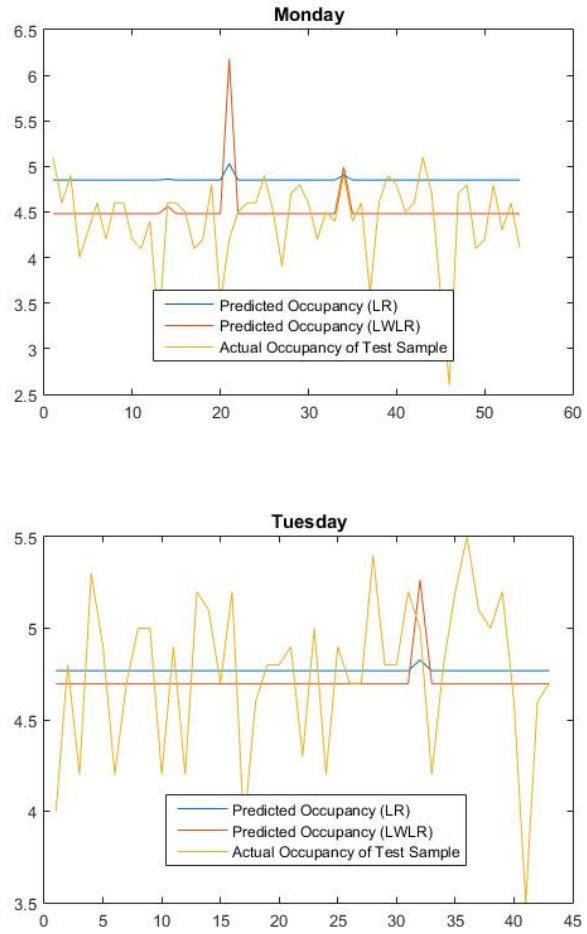

Fig. 4: Visualisation of results for fitting daily occupancy rates

IV. Detailed methodoloy and corresponding results

The methods described are implemented on hourly data in order to model the daily trends of lane occupancy. The corresponding results are described as well.

A. Linear Regression (LR)

First, we simply applied the most basic form of linear regression and solved it directly using the normal equations (a least squares approximation). With linear regression, several different approaches could be taken. We start with the naive approach of using only the features defining inputs in the dataset, and modeling them all as continuous variables. Finding this to be very inaccurate, we then introduce more features until we get a fairly good fit.

1) Approach 1: Treat all features as continuous: Only those features described in Table I and Table II are used for this method.

Each of the features is treated as a continuous variable. This means the algorithm fits θ such that $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \theta_3 \cdot x_3 = y$ where x_1, x_2, x_3 correspond to features 1,2 and 3 outlined in Table I, and y refers to the occupancy rate which is the output. As expected, this naive approach did not yield great results.

Sample of the fitted result is shown in Figure 5. The training error for our data set (calculated as root mean squared error) is **2.2650** with test error equal to **2.2770**.

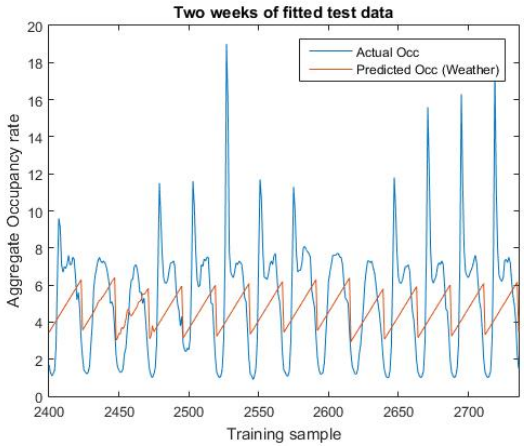

Fig. 5: Visualisation of Approach 1 for fitting hourly occupancy rates

2) Approach 2: Treating "Hour" and "Day" as categorical features: Treating different "time slots" as categorical features. By doing this we effectively introduce more "features" to which we will fit coefficients - ie. hours 0-23 are each a feature, and weekdays 1-7 are also features. The variables are then treated as a binary value - 1 if true, 0 if false, thus adding the related co-efficient if it is true and ignoring the term otherwise. By introducing more features, we are able to reduce the bias of our model and more closely approximate the actual data. A sample of the fitted result is shown in Figure 6. The training error for our data set (calculated as root mean squared error) is **1.3194** with test error equal to **1.3271**.

We notice that the predicted trend for Saturday and Sunday is just a scaled down version of a regularly weekday, but in reality those two days have a different trend due to the lack of a morning rush-hour peak. Hence, we introduce one more set of features to model this, in the third approach.

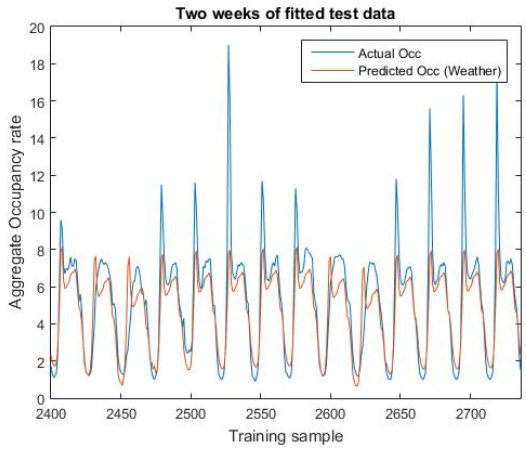

Fig. 6: Visualisation of Approach 2 for fitting hourly occupancy rates

3) Approach 3: Treating correlation between "Hour" and "Day" as a feature: A set of categorical features of **Day*Hour** is introduced to model the interaction between day of the week and the hour of the day. By introducing this feature, we teach it to recognize the difference between a weekday day and a weekend day.

A sample of the fitted result is shown in Figure 7, showing that the introduction of this feature successfully achieves what we wanted. The training error for our data set (calculated as root mean squared error) is **1.0356** with test error equal to **1.0667**, showing the model fits even better. However, to ensure that we aren't overfitting (the number of features significantly increases in this case), cross-validation of the data is conducted over randomized training and test sets.

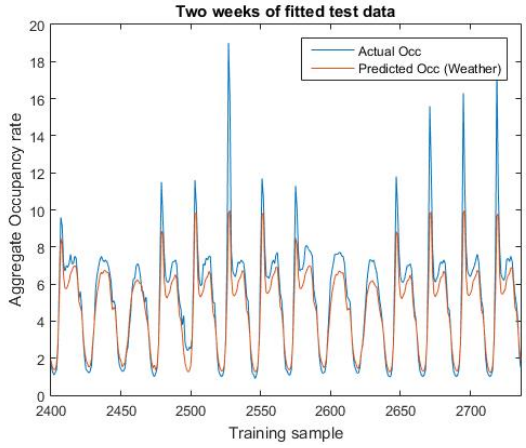

Fig. 7: Visualisation of Approach 3 for fitting hourly occupancy rates

4) Cross-validation of various approaches: Cross validation is conducted simply by splitting the data and using 70% for training, 30% for testing.

A summary of the root mean squared fitting error achieved with the three different approaches to linear regression are shown in Table III. These numbers are generated from a simple 70-30 split of the training set.

From this is can be seen that the third approach performs better in training error as well as in cross validation, hence it is the hypothesis which should minimize the generalized error across all other data sets.

B. Functional regression (FR)

For functional regression, we narrow our focus to predicting Tuesday traffic based on the preceding Monday. To perform the regression, we find the 7 closest training Monday time series to the Monday time series of the test example, based on the sum of squared error. We then look at the Tuesday time series corresponding to those 7 Mondays. We drop the time series with the highest individual value and the time series with the lowest individual value, because this should reduce the effect of outliers on our prediction. We take the 5 remaining Tuesday time series and take a weighted average, where the weights are based on how close the training Mondays are to the test Monday. Closer Mondays have a larger weight on their Tuesday data.

The graphs in Figure 8 and Figure 9 show examples of Tuesday predictions compared to the actual Tuesday time series. The shape of the predictions closely match the shape of the actual data in these examples.

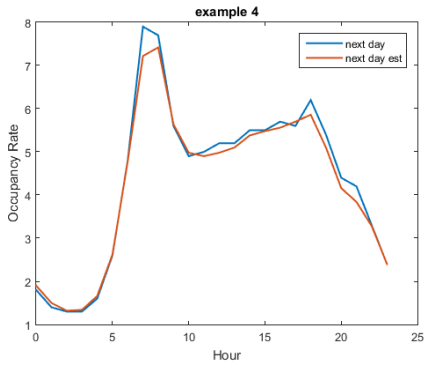

Fig. 8: Example 4 for estimated occupancy rate v.s. actual occupancy rate

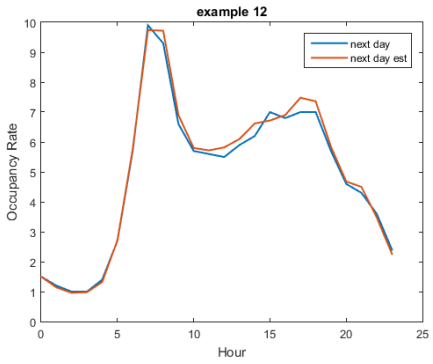

Fig. 9: Example 12 for estimated occupancy rate v.s. actual occupancy rate

To calculate the test and training error, we found the root mean squared error on an hourly basis. Although our functional regression prediction predicts an entire day at a time, calculating the error for each hour allows us to directly compare this method to the linear regression methods. As seen in Table III, functional regression produced more accurate results than our linear regression methods, even without using any weather data. The training error, however, is higher than the test error, because whenever the training example itself is treated as an outlier, it will be dropped from the set of nearest neighbors, reducing the accuracy.

TABLE III: Summary

	LR1	LR2	LR3	FR
Training Error (RMSE)	2.2650	1.3194	1.0356	1.0326
Test Error (RMSE)	2.2770	1.3271	1.0667	0.7549

C. Aside: Investigating the effect of weather on traffic

During this process of finding a model, we also check the significance of weather data to the accuracy of the model. By fitting linear regression models with and without weather as feature, it is found that weather data has little impact on the occupancy rate of the particular location we are analyzing on the I280 freeway.

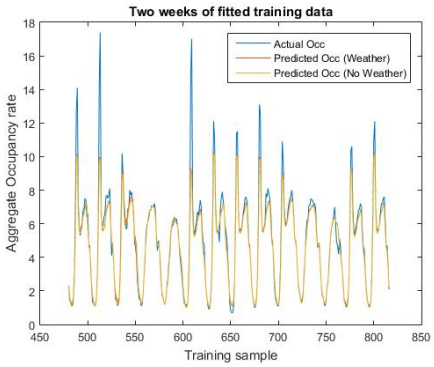

Fig. 10: Showing the non-effect of weather data

V. Conclusion

It is shown that Hour, Day, and Hour*Day are the most relevant features for linear regression whereas temperature and precipitation make little difference for predicting the aggregate occupancy rate. The superior performance of functional regression, which doesn't use weather data, provides more evidence that weather conditions do not affect occupancy rates as much as we initially expected.

Functional regression is fairly accurate, indicating that traffic patterns are regular enough that tomorrow's traffic can be predicted based on today's traffic. However, there were several outliers that initially decreased the accuracy of functional regression. Eliminating these outliers in training data greatly improves the test error. It's difficult to determine if these outliers are caused by sensor glitches or legitimate extreme traffic variations. Determining the cause of these outliers would be an important next step in this project, so that we could either get better data or add more features to include these traffic variations.

VI. Future Work

As mentioned in the conclusion, determining the source of outliers would be the next step in this project. If we are able to add features that model events like traffic accidents or clean our data so that it better represents reality, we should get better results. Once we have improved results, we would take data from many more sensors so that we could build a model that predicts traffic in an entire region instead of just at one point. This extra sensor data would both provide more useful predictions and add redundancy which could improve accuracy.

References

[1] Billy M. Williams, M.ASCE1 and Lester A. Hoel, $F.ASCE^2$, Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results

[2] Brockwell, P. J., and Davis, R. A. 1996!. Introduction to time series and forecasting, Springer, New York

[3] L. L. Ojeda, A. Y. Kibangou and C. C. de Wit, "Adaptive Kalman filtering for multi-step ahead traffic flow prediction," 2013 American Control Conference, Washington, DC, 2013, pp. 4724-4729

[4] S. H. Hosseini, B. Moshiri, A. Rahimi-Kian and B. N. Araabi, "Short-term traffic flow forecasting by mutual information and artificial neural networks," Industrial Technology (ICIT), 2012 IEEE International Conference on, Athens, 2012, pp. 1136-1141

[5] Ryan Jay Herring, Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning

[6] Advanced Data Analytics in Transport Machine Learning Perspective, https://research.csiro.au/data61/advanced-data-analytics-in-transport-machine-learning-perspective/

[7] A. Moussavi-Khalkhali, M. Jamshidi and L. B. E. Chair, "Leveraging Machine Learning Algorithms to Perform Online and Offline Highway Traffic Flow Predictions," Machine Learning and Applications (ICMLA), 2014 13th International Conference on, Detroit, MI, 2014, pp. 419-423.

[8] National Centers for Environmental Information - https://www.ncdc.noaa.gov/cdo-web/datatools/lcd

[9] Marco Cuturi, "Fast Global Alignment Kernels"

[10] California Department of Transportation - http://pems.dot.ca.gov/