Travel Time Prediction Problem RTA Freeway

27 April 2011

Authors:
Alex Zhicharevich
Yaron Margalit

1 Introduction

Imagine that you are driving on the freeway at 90 kph and trying to figure out on what routes you should drive on the next 15-30 minutes in order to get to your destination on time.

Forecasting travel times helps improve road safety and efficiency. Accurate predictions of the traffic management systems helps travelers to optimize their traveling such as bypassing congested segments of the network, change departure times and/or destination, whenever appropriate. Besides impacts and benefits at the traveler level, such decisions are likely to affect the potential demand at various points on the network and provide opportunities for better utilization of the existing infrastructure capacity, reduce congestion in network and consequently in reduced costs.

Forecasting provide the traffic state which the travelers will experience when they will use the traffic system. A proactive system is required to build the forecasting of the future state using the present and past traffic state information. Different types of data collection information can be used to asses the traffic state .We used fixed loop detectors which provide the traffic information at the specific location .The proposed methods uses the detector data for the predication of the travel time.

In this project, we will refer to the traffic prediction problem and investigate how this can be potential being solved in Australia Sydney's M4 freeway.

Outline. The rest of the project is organized as follows. Section 2 we formally define the traffic prediction problem, and give some initiatives that led us to the proposed methods. Section 3 presents the solution approaches including: History evaluation prediction and "random walk" evaluation. Section 4 presents the test applications built from the former approaches and experimental results on the Australia Sydney's M4 freeway. We end with some conclusion remarks in Section 5.

2 Preliminaries

2.1 The Problem

In the traffic prediction problem that we are trying to solve, we are given a freeway that was cut into a series of n segments $(R_1, ..., R_n)$ not necessarily of equal length. For each segment, the measurements of the travel time are given in intervals of 3 minutes, i.e. each day consists of 1440 time measurements for a certain segment (start of 00:00,00:03,...).

The problem requires to forecast each travel time on the freeway for 15 minutes, 30 minutes, 45 minutes, one hour, 90 minutes, two hours, six hours, 12 hours, 18 hours and 24 hours after the cut-off time x for each freeway segment-route (prediction time horizon). History related of the time measurements for each freeway segment of the last few months before the cut-off are given.

The target is to minimize the RMSE (Root mean square deviation) error. Where having an estimator $\hat{\theta}$ with respect to the estimated parameter θ we try to minimize:

$$RMSE(\hat{\theta}) = \sqrt{E((\hat{\theta} - \theta)^2)}$$

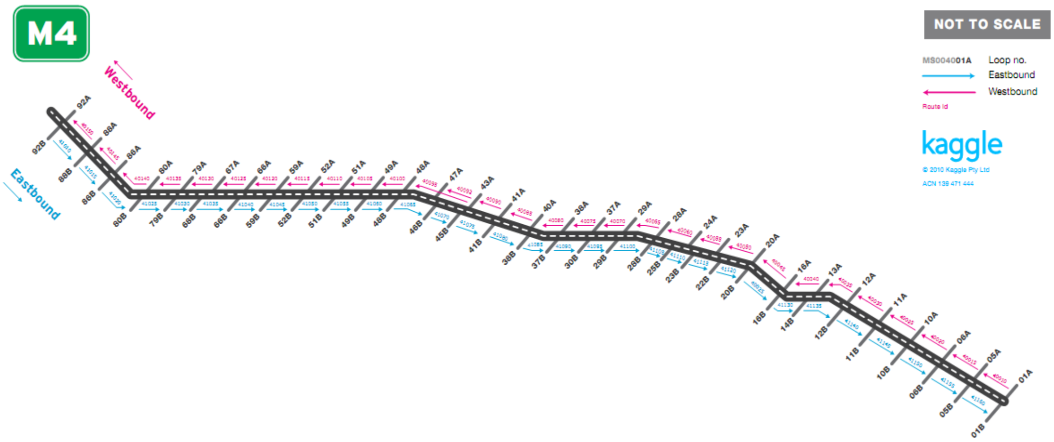

Figure 1: M4 topology map.

2.2 The Data - Sydney freeway

The Sydney freeway was cut to 61 segments, not necessarily of equal length, and some sensors on the freeway measure the travel time (in deciseconds) on each segment. About half of the segment are eastbound and the other half is westbound, and the topology of the segments on the freeway is given (by map - Figure 1). The lengths of each segment are also given in meters.

The main data consists of series of sensor measurements. For each segment (route) the measurement of the travel time is given in intervals of 3 minutes. The exact date and time of the measurement is given, and it is measured simultaneously on all routes. History data of the last few months of the freeway was included. In order to test and check our work, we had a test set that included 29 cut off times.

In most applications that we created, we had separate predictors for each individual route and for each prediction time horizon (i.e. 15 minutes ahead, 30 minutes ahead, etc). So, the solution required for each cut-off to build 6110=610 (61 routes x 10 prediction time horizons) models.

2.3 Initiatives:

Obviously, a statistical model should be established. Upon building such a model we would like to understand what are the variables that should be considered for predicting the traffic situation at some point in time.

It looks natural that the traffic situation has some cyclic nature. Certainly, each day has some peak and off peak hours, but they may be different each day and the congestion's severity may vary from one day to the other. Same as for daily cycles, a weekly cycles also should be taken into account, since we understand that difference between the transportation nature between weekday and weekends (different traffic states as shown in Figure 2). There are also anomalies in the historical data like holiday, special events, election days etc. There is also significance to the year periods, since driving speed and accidence rate may change between summer and winter. We need to establish some method to take all historical data into account and to score the relevance of each historical observation. There is also a trade-off between the amount of the available and relevant data to the accuracy and robustness of the model we would like to build. A model for each day

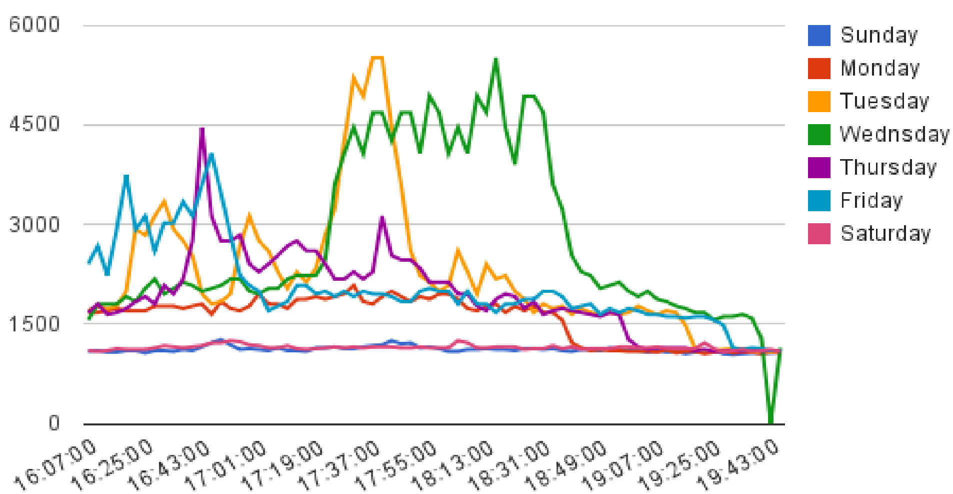

Figure 2: A specific route peak travel time distribution (time between 16:00-20:00) on a random week by weekdays.

hour and weekday may leave us with too little relevant data to establish a model, while trying to avoid some differences mentioned above may lead to an inaccurate model.

Beside a purely statistical model some pattern detection method may be also helpful. It looks possible to recognize major car accidents by some changes in the congestion rates. If such pattern recognition method could be established we could give more accurate predictions based on previous observations of such anomalies that can be caused by accidents, road works, extreme weather conditions and so on. This type of method should also consider the latest observations and try to predict what is the trend of the congestion.

We would also like to examine the correlation between traffic states on neighboring routes. Intuitively, if there is a congestion on some part of the highway, sometime later it is expected to arrive to the next part. On the other hand we understand that some parts are more central then other. It is also not clear that a model that works well on one route is also the most suitable for all of them or should a different model should be trained for each one of them.

In the next sections, we present some of our approaches to those questions, and statistical data that tries to explain why some of the approaches are better then the others. The main algorithms that we taken in our project consists of: Batch learning, Decision trees (including regression), time series analysis (such as ARIMA), pattern match and random walk hypothesis implementations.

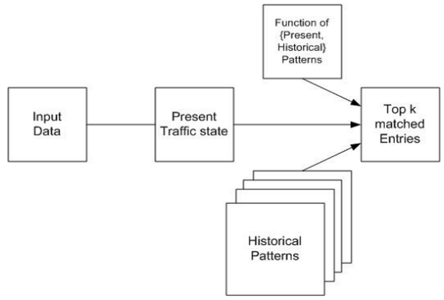

Figure 3: History model.

3 Solution Outline

3.1 Historical evaluation prediction

The hypothesis of the historical evaluation prediction relies on the fact that traffic conditions are recurrent in nature. If we assume that present traffic state is defined in a way which properly relates to the travel time, we can search in the history a similar traffic states that can be used to extrapolate the present traffic state to predict the travel time. The outline of the historical evaluation model can be interpreted as shown on Figure 3.

Referring to the Figure 3, the proposed approach select a check function. Function could be some sort of a similar function or a standard operation. Given the present traffic state and historical database (history traffic states), the algorithm uses the check function. One example of usage would be that the function is a similar function. In that case, the algorithm search for similar instances in the historical database and returns the top k most similar patterns.

In this section we will refer to some different implementations of the proposed model.

3.1.1 Short time history

The first implementation of the historical evaluation prediction is very basic and intuitive. We assume that the predicted value only depends on values that are from the same time in last previous days (60 days). The function returns only one entry such that:

$$Value_{route,date,time} = f(Value_{route,date-1,time}, Value_{route,date-2,time}, , Value_{route,0,time})$$

The trivial f would be just taking the average of all previous seen values. For every route we computed the average of all previous measurements where we take only weekend measurements or non weekend measurements, depending on the prediction date. A simple median of all values and root of sum of squares were also tried and did not show superior results.

3.1.2 Moving weighted average

As showed one example on Figure 4, estimation traffic state from recent values better predict the current value than earlier ones. This is somehow seems more realistic, since recent days may present similar weather conditions and better reflect the road conditions.

The average function (from previous section) was replaced by some more complex model known as weighted average and weighted moving average [8]. Weighted average assumes that more recent

n	Time		
	18:22	11:19	15:25
14	343.05	14.2	226.06
28	342.3	14.35	228.58
42	334.52	15.09	239.5
56	340.62	16.04	244.56
70	365.89	17.24	259.41

Figure 4: RMSE of prediction example by averaging the last n day in the same time exactly (on some route on several times in the same day).

Window	RMSE
6 mins	243.4
12 mins	242.5
18 mins	242.0

Figure 5: RMSE of prediction by averaging the last 7 days with n minutes window (on the same time on the same route).

values better predict the current value then earlier values, so values that were measured later receive more weight in the averaging process. The moving weighted average takes the weighted average only on a moving window of the last n measurements days. The weight function and the window size need to be estimated empirically. Obviously, for weekend that have fewer values, we need to take a bigger window days. Eventually a window of 40 and 60 days was tried for non weekends and 80 days window for weekends. In addition, since we take only a window of latest data, the number of measurements that we average becomes smaller, so the average is more vulnerable to noise. To make it more robust we can take for each previous day not only the measurement in the exact same time but in some time frame of close times. The trade-off here is between recent values on close times of day and older values on exactly the same time (See Figure 5).

As for weighting the values, the weighted moving average model states that recent values should be given more weight.

1. **Linear weighting** - for predicting time t $weight(t-i) = \frac{a}{i}$ for some a
2. **Exponential weighting** - for predicting time t $weight(t-i) = e^{(-ai)}$ for some a.
3. **Discrete weighting** - for predicting time t $weight(t-i) = \frac{1}{(i\ div\ a)}$ for some a.

Exponential, linear and discrete weighting were tried and the discrete weighting for $a = 7$ was superior, meaning for some prediction the measurements made in the last week were weighted the highest, the week before less etc.

3.1.3 ARIMA

Classical ARIMA (Auto regressive integrated moving average) is widely used in time series analysis and particularly in traffic prediction [1, 2]. It is parametrized by parameters p,d,q where:

1. p is the number of auto regression terms.

2. d is the number of difference terms.

3. q is the number of moving - average terms.

In this case, ARIMA was used as a function of time series check (same usage as the average function on previous section) in order to predict future points in this series. One can see that the model contained no difference terms. The results of many differently configured ARIMA models were compared. A range of values of both the auto-regression parameter p and the moving-average parameter q were experimented with. Best results were picked as ARIMA(1,1,1).

3.1.4 RMSE Pattern Match

So far, most implementations were taken function f as an operational function. In this section, we will use a similar pattern match function. The basic aim of the pattern matching procedure is to find the most similar historical pattern. RMSE difference between the current time traffic pattern and the historical traffic pattern is used as a criterion for finding similarities between the traffic patterns [2, 3]. The historical traffic pattern that has the minimum RMSE different is regarded as the most similar pattern.

In order to commit the pattern match procedure we have some challenges that we need to take into account. The size window of the pattern match need to be decided, that is, the time frame before the "cut-off" (for example having a time frame of one hour). Another issue is searching time frames over the historical data. Ideally, we would want to search the whole historical pattern data, but with limited time and computationally power, we will add the assumption of that similar traffic recur in a tight time frame. Such as, similar traffic state for 6pm can be searched from 5pm to 7pm. Time patterns were searched for closed patterns. Last issue, regarding the output of the method was how many matched entries we would like to obtain, empirically we saw that having the most k=3 matched entries (minimum RMSE different) and average over the results gave the best result.

3.2 Random Walk Evaluation prediction

The second approach was the random walk evaluation [4], meaning that the value of the measurement depends mostly on the last values measured on the same route. For example, if we need to evaluate the value on some route at 15:00 and we have the value of 14:00, it will make sense to use it and not only the history measurements in 15:00. In the following sections we will relate on how to catch the recent trends and the growth rate using the last measurements.

3.2.1 Simple Random walk

The simplest approach is the pure random walk approach. If the values behave like a random walk, the optimal strategy is to predict the last value seen. Given the last measurable time before the "cut-off" predict the time travel the same as the last measurable time. As shown in Figure 6 the random walk accuracy (RMSE different) linear reduces as we move away from the last measurement. We will relate more about that in Section 4.

3.2.2 Trend estimation

The assumption that traffic congestions behave as a random walk is rather naive, since it is obviously cyclic in some way. It is clear to us that if a measurement is taken at some hour of the day, the next

15 mins	136.2431
30 mins	164.9695
45 mins	183.0129
60 mins	197.1332
90 min	223.5017
120 mins	243.5621

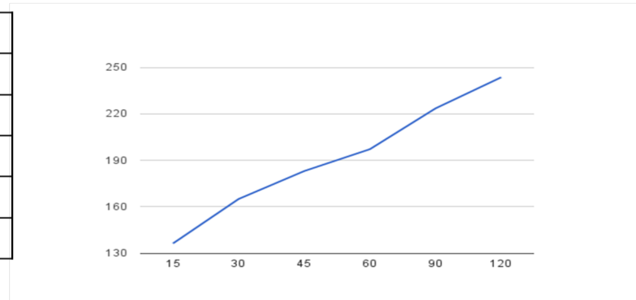

Figure 6: The RMSE of random walk prediction according to to time difference from last measurement (on a single route).

values are not randomly walking from this measurement. For example, if our last measurement was at peak hour and we need to predict an off-peak hour value we can be pretty sure that the value is going to drop and we can estimate by how. So to improve the random walk approach we want to use some extra knowledge about the data. Since we have the measurement in a certain time distance from the prediction time, we can estimate how the value changes from the last measurement to the prediction time, i.e. to find some trend estimation [5].

The estimation is done by computing the average ratio between measurements on the same route on those times. For example if we need to predict a value $V_{n,14:00}$ at 14:00 and we have a measurement $V_{n,13:00}$ from 13:00, we can estimate $P = V * Average_{i=1...n}(\frac{V_{i,14:00}}{V_{i,13:00}})$ where $V_{i,t}$ is the measurement at day i at time t. The adjustment showed a significant performance improvement over the simple random walk. Obviously, the ratio can be also estimated using weighted moving average instead of a simple average but this approach was not tried.

Another approach was using average difference between the times instead of the ratio. Meaning $P = V + Average_{i=1...n}(V_{i,14:00} - V_{i,13:00})$. Figure 7 shows that the performance of this estimation is slightly lower then the ratio estimation, but it might work on other scenarios.

Another trend estimation that we used as explanatory variables to find a specific relation to the predicted travel time with decision trees: 4.1.4

1. Growth rate over the last two most recent times: $X_{t-1} - X_{t-2}$
2. Growth rate over the average of the last two most recent times and preceding them: $(X_{t-1} + X_{t-2})/2 - (X_{t-3} + X_{t-4})/2$. This trend is more robust.

3.2.3 Recent Value Weighting

In the historical approach we used weighting of the historical values in order to improve the prediction over simple average. It looks like if there is no prior knowledge then giving more weights to recent values is the natural approach. But in our case, we can use the recent value measured on the route as a prior knowledge and use it to make smarter weighting. The assumption is that we should give more weights to previous days that presented similar traffic situation to the day we want to predict.

For example, if we want to predict the value in 15:00 on some day, and at 14:00 we measured on the same day on the same route a value X. We would like to give higher weights to days that measured values close to X on 14:00. This looks natural since if X is some peak, we would like to

Trend estimation	15:00-16:00 RMSE	21:00-21:30RMSE
No trend	686.4396	137.4367
Additive trend	646.3047	136.6781
Ratio Trend	644.9067	136.348
Weighted Ratio Trend	642.813	136.3224

Figure 7: Comparison of trend estimation techniques. On a single route we took all measurements over 2 points of time, and used the first 50 measurements to estimate average ratio and difference. We estimate the RMSE on predicting the second points using trend estimation and the first point.

give higher values to days that had peaks on 14:00. This sort of weighting hold also for historical estimation and for trend estimation explained in the previous paragraph.

The weighting should be done by some formula that gives closer values higher weights, so it should look like this: if the last measurement at the prediction day was taken at hour X, then: $Weight(day_I) = \frac{a}{|Day_n(X) - Day_i(X)|}$ for some a.

Unlike the regular historical weighting, this weighting is accurate only for predictions that lie within some time window from the last measurement, as all random walk methods. So there should be some mechanism that chooses when to use random walks weighting and when to use other time-based models. We chose to use this weighting only on predictions that lie within 2 hours from the last measurement, more details regarding this in Section 4.1.1.

Weighting by recent value is only one quite simple type of weighting according to similarity between traffic situation over days. It looks like the weighting technique is of less significance then the similarity function. We defined the similarity function $F(day_I, day_J)$ as the absolute value of the distance between their traffic conditions at a specific time point. There can be a variety of such functions and it looks like a pattern recognition problem. Another approach we tried is to take the 5 last measurements of Day_i and Day_j and set F as 1/RMSE of those values. This approach showed poorer results then just last value difference, but some parameter adjustment can improve it.

3.2.4 Neighboring Routes

One more observation we had is the neighboring routes effect [6]. If the most recent data for route B show longer travel times, and route B is next to route A, it's likely that, as traffic moves along to route A, travel times on route A will increase. We took neighbors into account, specific relation to predicted travel time was part of the decision trees in 4.1.4.

4 Applications, Experiments and Results

4.1 Applications

In the previous section, we outlined two main approaches in order to have a prediction. Obviously, it seems that taking only one approach each time won't give us good results. As we move away from the last measurement (the "cut-off") performance of the random walk evaluation approach will drop. This is due to fact that the correlation to the traffic state of the last measurement is reducing. The same holds for the historical evaluation approach: the closer we are to the cut-off the historical approach is less relevant. Therefore some weighting formula should be established for combining the recent value and the historical value calculated by the previous section. We specify here some applications that combine those approaches by some sort of weighting, pattern match or part of regression to estimate effects.

4.1.1 Weighting history with random walk evaluation

The weighting between the history predict should be more significant in predicting values measured long after the last measurement. Here we want to define a formula for the weight of random-walk evaluation prediction (the weight is between 0 and 1). Denote the time difference (in minutes) between the last measurement time and prediction time by d. So we want $weight_{random-walk}$t depend on d. Here can also give exponential, linear and distinct weighting, when we can see in Figure 6 an example for the random walk implementation. For $d = 15$ and $d = 30$ we get very good results and for $d > 120$ the results are almost irrelevant. We chose the linear weighting so $weight_{random-walk}(d) = min(1, \frac{a}{d})$ where $a = 20$.

4.1.2 Pattern match application

The pattern match application is based on solution outline in 3.1.4. We tried a complex pattern matching technique, by searching all previous measurements for traffic pattern similar to the prediction day. We took pattern match results and combined it with historical values using the same approach as in the previous section.

4.1.3 PAC style historical evaluation

Here we used a rather simplistic statistical model and enhanced it. We took a moving window of 60 days back, and for each prediction established a set of hypothesis: mean, average and root of sum of squares of all previously seen values. The values are picked from a time window that varies from exactly the same day time up to 30 minutes window in intervals of 6 minutes. So we had 18 hypothesis and for each prediction we selected the best hypothesis that predicts the training set. Training set included last 10 days on the same route. The chosen hypothesis which minimized the RMSE error is then used to predict the actual prediction.

4.1.4 Random forest evaluation

The last approach was a statistic learning approach: for each prediction we extract a set of explanatory variables which are the time of day, the weekday, the date, the recent value observed, the growth rate (as mentioned in trend estimation section) on the same route and neighboring routes values. We then used the last month of the observations as a training set to train an ensemble of regression trees for these features [7]. In order to avoid noise and finding the best split we had a number of decision trees. Each decision tree had a random subset of the explanatory variables and training data. Prediction is done by taking the average of the multiple regression decision trees. A model is established for each route, for each time distance from the cut-off point. After the model is established, the prediction is made using the ensemble of regression trees trained on the model.

Application	All	per Direction	
		Westbound	Eastbound
ShortTimeHistory	212.99	241.43	181.26
Weighted moving average (WMA)	214.62	243.79	182
Random walk	418.53	485.67	341.21
RMSE pattern match	215.02	244.56	181.93
WMA + random walk	205.92	231.8	177.32
WMA + trend estimation	203.98	228.49	177.06
WMA + recent value weighting	201.77	226.41	174.66
ARIMA + trend estimation	206.23	231.46	178.45
PAC model	219.66	248.53	187.55
TreeBagger	207.14	232.58	179.11

Figure 8: The RMSE results of the algorithms on the entire test set are presented.

Application	Rush hour time	Not rush hour time
ShortTimeHistory	326.92	158.27
Weighted moving average	330.62	158.66
Random walk	582.47	347.81
RMSE pattern match	331.03	159.1
Weighted moving average + random walk	323.15	148.05
Weighted moving average + trend estimation	319.94	146.78
Weighted moving average + recent value weighting	316.35	145.28
ARIMA + trend estimation	320.12	150.76
PAC model	345.32	157.49
TreeBagger	326.78	147.68

Figure 9: The RMSE results partition of rush hours and off rush hours traffic.

4.2 Experiments

Referring to Figure 8, We can see that weighted moving average combined with recent value weighting presents the best results. It can be seen that weighting history with random walk evaluation performance significantly improve results than running each approach separately.

We can also see that although the eastbound and westbound routes present very different traffic states, an algorithm that worked well on eastbound routes worked well on westbound routes as well. We can deduce that a single model can be built and work well on all routes, and there is no need to apply different algorithms to different routes.

From figure 9 we can see that rush hour traffic is significantly harder to predict than other hours. It is expected since rush hours have generally higher values, but also they present more unexpected situations then other hours. This fact can be used to maybe take more measurements on rush hours, since they are the bottleneck for prediction and more interested to our general problem.

From figure 10 we see that random walk methods present good performance on near cut-off times, while historical methods have pretty stable behavior over all intervals from cut-off. This explains the superior performance of combining random walk and historical approaches

Regarding the computation view of the applications, one of the major requirements from a prediction is to have fast results. This is since there is no relevance to application that release a prediction after the car already in the middle of the traffic jam. In most of the applications we created, computation time of the training took time between several minutes to a few hours. But, we created such a model, the prediction itself didn't take more than a few seconds. For example, relating to the regression decision trees the training time had a significant implication over the test

Application	Cut-off									
	15 min.	30 min.	45 min.	60 min.	90 min.	2 hours	6 hours	12 hours	18 hours	24 hours
ShortTimeHistory	246.5719	336.5032	286.4738	201.4246	181.9411	198.8278	127.859	101.2818	73.16355	228.9073
Weighted moving average	248.8659	336.5866	286.3046	199.1159	177.2489	198.4379	133.2566	98.15182	75.48095	244.7984
Random walk	191.4835	336.2978	322.5802	290.4663	331.9988	369.6918	360.5354	355.9583	326.8582	896.1811
RMSEpattern match	247.9046	336.4099	287.1365	201.4957	178.5795	199.2061	133.4345	98.52903	75.99532	244.5982
Weighted moving average + random walk	189.487	318.1539	277.0591	196.0549	176.9506	201.7491	134.8533	99.11292	79.40497	255.7396
Weighted moving average + trend estimation	186.602	312.8675	277.6036	194.2486	174.4443	196.4682	134.304	99.11292	79.40497	255.7396
Weighted moving average + recent value weighting	181.0866	311.1127	276.6445	192.3267	176.6534	196.2613	132.6686	97.54494	76.69375	247.6223
ARIMA + trend estimation	186.9742	312.1187	282.1586	199.5348	180.2749	195.6373	157.9521	98.61348	72.47231	250.6149
PAC model	249.8431	347.4442	290.5155	198.3342	176.8258	209.3572	134.2396	105.6029	69.55026	257.9945
TreeBagger	197.84	319.26	277.6	196.84	176.7	203.09	134.85	99.11	79.45	255.73

Figure 10: The RMSE results partitioned to prediction time after cut off.

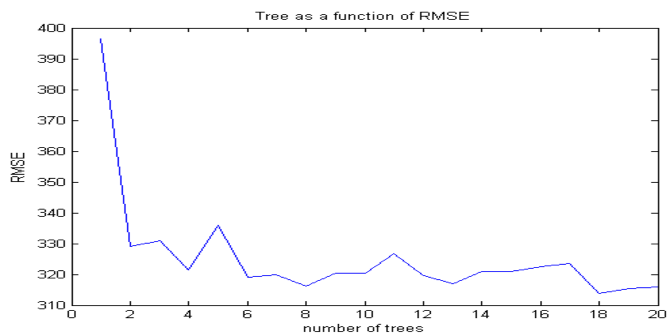

Figure 11: The average RMSE results depends on the number of decision trees for a specific route, 6 hours from cut off. Each tree decision takes around one second.

results. Relating to Figure 11, after training using 20 trees, we obtain a much lower RMSE error (than using only a few decision trees). After having such a model, the prediction time takes a few seconds.

5 Conclusion and Discussion

Predicting traffic conditions is an important task for many applications. The widespread of modern sensors allow us to produce massive data sets for statistical learning of the problem. We have shown various methods for such predictions, some of them may be applicable to other time-series analysis.

From our experiments, it looks like there is some limit on what can be achieved using purely time-series analysis. Effective models like ARIMA did not show to be superior over more simple averaging models, and although this approach looks like the most robust one it should be enhanced by some additional features to present good results. It is also seems that the parameter adjustment of the methods does not increase accuracy.

More work can be done on estimation the quality of the output predictor based on data, i.e. the validation procedure for model selection. Since the data is scarce and we do not want to "waste" data on validation. The cross validation technique could be used for model selection (or parameter tuning), it is often works very well so it can be very interesting to see this relating in our applications.

The most promising technique as we see it is to produce some pattern recognition method to try and find past patterns in the data and match them to the new data to be predicted. We only tried several methods in this direction and it seems more effort or some other ideas may lead to a significant improvement.

Looking into other related work on traffic prediction problem, we didn't find any methods that overcame the naive methods by a significant gap. This brings up the question on whether traffic congestions can be effectively learned using statistic models.

References

[1] M. Voort, M. Dougherty, S. Watson. Combining Kohonen maps with ARIMA time series models to forecast trafic flow. Transportation Research Part C: Emerging Technologies 1996;

[2] B. Williams, L. Hoel. Modeling and Forecasting Vehicular Traffic Flow As a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. Journal of Transportation Engineering, Vol. 129, No. 6, 2003, pp. 664-672.

[3] J. Van Lint. Reliable Travel Time Prediction for Freeways, Delft University Press, Delft, The Netherlands 2004.

[4] Wikipedia: http://en.wikipedia.org/wiki/Random_walk_hypothesis

[5] M. Bianchi, M. Boyle, D. Hollingsworth. A comparison of methods for trend estimation, Applied Economics Letters, 6(2): 103-109, 1999.

[6] Y. Kamarianakis, P. Prastacos, P. Bivariate. Space Time Modeling of Traffic Flow. Computers and Geosciences, 2005.

[7] H. Chem, X. Yao. Ensemble Regression Trees for Time Series Predicitions. IEEE Congress on, 2007.

[8] Wikipedia: http://http://en.wikipedia.org/wiki/Moving_average