The Impacts on Fuel Consumption: A Data Mining-Based Analysis

Akash Iyengar¹ Karthigainathan.M², Kalpana.G³

¹SRM University, Department of computer science and Engineering,
²SRM University, Department of computer science and Engineering, Assistant Professor,
³SRM University, Department of computer science and Engineering,

Abstract- This paper suggests different driving techniques based on the results of an applied research on the eco-driving domain, supplemented by a huge data set produced from Delhi’s transport system. The data set is based on events automatically extracted from the control area network and enriched with information like GPS coordinates, weather and road data. We use online analytical processing (OLAP) and knowledge discovery (KD) techniques which handles the high volume of data and to determine the major factors that determine the average fuel consumption, and assist in classifying the drivers based on their driving efficiency. Our findings leads to the introduction of simple practices, such as optimal clutch, engine rotation, engine running in idle state and traffic precautions, can reduce fuel consumption on average from 3 to 5/l/100 km, meaning a saving of thousands of litres of petrol per day. With the availability of traffic through various traffic sensors, a lot of research effort has been involved in developing traffic prediction techniques, which in turn improve route navigation, traffic minimisation etc. One key boon in traffic prediction is the reliability on prediction models that are constructed on the basis of historical data applied in real-time traffic situations, which may differ from that of the historical data and has a tendency to change over a period of time. We aim in obtaining and proving both short-term and long-term performance bounds for our online algorithm. The proposed algorithm also works effectively in scenarios where the realized traffic are missing or are available with a delay. In this paper we used a novel online framework that could learn from the current traffic situation in real-time and predict the future traffic by matching the current situation which will be useful for the effective prediction of fuel consumption and a suitable driving scheme.

Keywords – eco-driving, fuel efficiency, OLAP, KD, fuel consumption, traffic prediction big data, spatiotemporal, online learning.

1. Introduction
Evaluating driver performance and promoting energy efficient driving has received scarce attention from the research fraternity. This is due to the difficulty in analyzing driving skills of an individual. The driver controls the speed, acceleration, braking, engine rotation speed, the gear engaged, the position of the vehicle on the street and the path he chose to drive in an environment characterized by certain traffic conditions, itinerary, load, etc. Different driving styles and different traffic conditions result in different fuel consumption levels, thus related to driving efficiency. There are various external factors which also determine the fuel consumption levels. For example, levels of fuel consumption in a public bus are strictly linked to the number of stops made per itinerary unit. The number of stops is a parameter that is not controllable by the driver, but on traffic, the bus route, or the number of passengers. One of the most efficient approaches to evaluate driver performance is to register a set of events (parameters) read from the CAN bus, which stores messages from all driving events on an on-board recorder, from where data is retrieved and stored in a database for subsequent analysis. Due to analytical needs and the huge data generated from this approach, this is a field where the application of online analytical processing (OLAP) and knowledge discovery (KD) techniques is needed. KD in databases has been attracting a significant amount of interest from both research and industry. We apply OLAP and KD techniques to deal with the high volume of data and extract the associated knowledge into big data scenarios i.e. the impact factors (IFs) that most influence the average fuel consumption (AFC) in litres per 100 km. Traffic congestion causes tremendous wastage in terms of both time and energy. According to a recent report from the Texas Transportation Institute in 2007, 439 metropolitan areas experienced 4.2 billion vehiclehours of delay, which is equivalent to 2.8 billion gallons in wasted fuel and $87.2 billion in lost productivity sensor instrumentations of road networks in major cities as well as the vast availability of auxiliary commodity sensors from which traffic information can be derived (e.g., CCTV cameras, GPS devices), a large volume of real-time and historical traffic data at very high spatial and temporal resolutions have become available. The traffic datasets can be used to predict
traffic congestion, which in turn enables drivers to avoid congested areas (e.g., through intelligent navigation systems), policy makers to decide about changes to traffic regulations (e.g., replace a carpool lane with a toll lane), urban planners to design better pathways (e.g., adding an extra lane) and civil engineers to plan better for construction zones (e.g., how a short-term construction would impact traffic).

**Process-2:** The data pre-processing process involves the identification data of the used, data manipulation towards a pre-defined class, performed by Tactic experts as well as the identification of outliers. Inconsistent data was also removed in an outlier’s implementation approach. This outlier process reduces the data in 20%, so after Process-2, we have 12M (completed) event records. Since we worked with 44 parameters, each outlier’s identification (in any of these 44 parameters) originated the removal of the complete data of that event.

**Process-3:** The OLAP process (detailed in Section IV) performed data analysis and helped to identify the variables with more impact in what concerned fuel savings.

**Process-4:** Finally, a data mining process was used to determine the impact factors (IFs) that most influenced fuel savings. We designed and built a decision support system that allows slicing and dicing by any selected dimension. Then, we applied Data Mining techniques to find the hidden patterns to allow the assessment of drivers, vehicles, routes, periods of the day and meteorological conditions in the selection of the most efficient entities in any given scenario. This system was developed with the help of Microsoft SQL Server 2008R2 with SSIS (SQL Server Integration Services), SSAS (Microsoft SQL Server Analysis Services) and SEMMA.

**The OLAP Process**
OLAP tools are designed to simplify and support interactive data analysis, but the goal of the KD approach is to automate this process, as much as possible, towards the identification of patterns. Pattern identification is based on fitting existing data to a model or commonly make any high-level description of a set of data. The KD process comprises many activities, namely data simplification from the outliers’ identification, pre-processing, and the search for patterns, knowledge identification, and refinement. All these activities should be repeated in several iterations. This means that KD is ahead of what is currently supported by most standard database systems. The platform used in this research has several OLAP features available to help analyze multidimensional data interactively, from multiple perspectives. In general, OLAP involves three analytical operations: (i) consolidation or roll-up; (ii) drill-down; and (iii) slicing and dicing.

**2. Work Methodology**

**Process-1:** The data collection process occurred between 2010 and 2012 (from Jan 1st 2010 until Dec 31st, 2012, with 1096 consecutive days). This process collected driving event data from Lisbon’s bus fleet and stored this information in a SQL database.

**3. The Data Mining Process**
The major objective with the data mining process was the analysis of fuel consumption per driver. The approach was complex due to the 30 dimensions involved. In the literature, we find...
several data mining approaches and algorithms, but because we had training data, we used the Naive Bayes (NB) Approach supported by the Microsoft platform (SQL Server 2008R2). This approach has a better performance with discrete Data.

1. The main idea is to divide the data population in subclasses with approximately the same number of events. An example of this is the average fuel con-assumption (AFC), by imposing it onto the DISCRETIZATION SSAS type in the training data set, with the input of five subclasses. This discretization process is based on an interactive maximized expectation algorithm, in order to divide training data into groups of similar population size. This technique was selected due to the presence of pronounced peaks, and because this method selects ranges of buckets to contain equal quantities of cases.

2. The division of data population based on percentage. We decided to divide each class into five subclasses, namely: on average, below average, above average, extreme above average and extreme below average. For example, the acceleration events (class Ac), captured from XtraN data, may have events ranging from 100 to 3500 (these numbers mean the number of times in 100 km that the accelerator pedal was used). The first subclass, AC1, ranges from 50 to 510, so the upper limit corresponds to 15% of 3400 (maximum value less minimum value). The second subclass, AC2, goes from 510 to 1190 (so the upper limit corresponds to 35% of 3400). The third subclass, AC3, goes from 1190 to 2210 (so the upper limit corresponds to 65% of 3400). The fourth subclass, AC4, goes from 2210 to 2890 (so the upper limit corresponds to 85% of 3400). And the last subclass, AC5, goes from 2890 to 3400.

A. Traffic Prediction
Several traffic prediction techniques have been studied in the past. The majority of these techniques focus on predicting traffic in typical conditions (e.g., morning rush hours) and more recently in the presence of accidents. Both qualitative and quantitative approaches have been used to measure the impact of an accident on road networks and various machine learning techniques have been applied to predict the typical traffic conditions and the impact of accidents, including Naive Bayesian classifier, Decision Tree classifier and Nearest Neighbour classifier. The main differences between our work and the existing studies on traffic prediction are: 1) All existing approaches for traffic prediction aim at predicting traffic in specific traffic situations, e.g., either typical conditions or when accidents occur. Instead, our scheme is applicable to all traffic situations and learns to match the current traffic situation to the best traffic prediction model, by exploiting spatiotemporal and other context similarity information.

2) All existing approaches used for traffic prediction deploy models learned offline (i.e., they rely on a priori training sessions) or they are retrained after long periods and thus, they cannot adapt to (learn from) dynamically changing traffic situations. Instead, our scheme is able to
dynamically adapt to the changing traffic situations on the fly and improve the traffic prediction over time as additional traffic data is received.

3) Most previous work is based on empirical studies and does not offer rigorous performance guarantees for traffic prediction. Instead, our scheme is able to provide both short-term and long-term performance bounds.

B. Ensemble Learning

Our framework builds a hybrid traffic predictor on top of a set of base predictors and thus, it appertains to the class of ensemble learning techniques. Traditional ensemble schemes for data analysis are mostly focused on analysing offline datasets; examples of these techniques include bagging and boosting. In the past decade much work has been done to develop online versions of such ensemble techniques. Another strand of literature on online ensemble learning is represented by prediction with expert advice and the weight update schemes. These algorithms assign weights to experts and make a final prediction by combining the experts’ predictions according to the weights. The weights are updated in a manner that may enable regret bounds to be derived. Most of these schemes develop multiplicative update rules. For example, the weighted majority algorithm in decreases the weights of the experts in the pool that disagree with the true label whenever the ensemble makes a mistake. Additive weight update is adopted in where the weights of the experts that predict correctly are increased by a certain amount. In weights of the experts are updated based on stochastic gradient descent. However, none of this work considers the context information when making the prediction (or equivalently, they consider that the context is the same in all time slots). We do consider context information and hence, our benchmark for regret analysis is much tougher. Specifically, in the existing work, the regret is defined with respect to the context-free benchmark in which the predictions are all made by the single best predictor ignoring context information. In our paper, the regret is defined with respect to the context-dependent benchmark in which the predictions are made by the best predictor conditional on each context. Given any context arrival process, the sum reward obtained by the context-dependent benchmark is greater than that by the context-free benchmark. Thus, even though existing weighted majority type algorithms can achieve a good (e.g., sub linear in time) regret bound against the context-free benchmark, the regret bound will not be sub linear in time when compared against the context-dependent benchmark. In contrast, our algorithm achieves a regret bound that is sub linear in time compared against the context-dependent benchmark, thereby providing both short-term and long-term performance guarantees. When there are several contexts, previous work provides regret bounds on average over all contexts while our work provides regret bounds on each context separately.

C. Contextual Multi-armed Bandits

When establishing the regret bound of the proposed algorithm, we adapted techniques from multi-armed bandit (MAB) problems since techniques used for ensemble learning problems, such as weighted majority type algorithms, lead to weak regret bounds for the considered contextual learning scenario. In our setting the prediction action does not have an explicit impact on reward realization and the learner can observe the realized rewards of all predictors. Hence, the considered problem is not an MAB problem and our algorithm is not an MAB algorithm. In our proposed algorithm, all time slots are equal in terms of the algorithm implementation and operation and there are no exploration or exploitation slots. In principle we could analyse all the time slots in the same way. However, this would lead to weak regret bounds. To get our strong regret bounds, we exploit the fact that we have stronger confidence bounds of the reward estimates in some slots than in others and hence, we use different ways to bind the learning loss in different slots. We divide slots in two types: type-1 slots represent slots for which we can have stronger confidence bounds of the reward estimates while type-2 slots represent slots for which we do not have such strong confidence bounds.

Fig 5 Architecture of learning algorithm

I. Algorithm description

First we introduce several useful concepts for describing the proposed algorithm: i) Context subspace. A context subspace C is a subspace of the entire context space, i.e. C ⊆ . In this paper, we will consider only context subspaces that are created by uniformly partitioning the context space on each dimension, which is enough to guarantee sub linear learning regrets. Thus, each context subspace is a D-dimensional hypercube with side
length being 2−l for some l. We call such a hypercube a level-l subspace. ii) Context space partition. A context space partition P is a set of non-overlapping context subspaces that cover the entire context space. Since our algorithm will adaptively partition the context space by adaptively removing subspaces from the partition and adding new subspaces into the partition, the context space partition is time-varying depending on the context arrival process of the traffic incidents. Initially, the context space partition includes only the entire context space, i.e. P0 = {}. iii) Active context subspace. A context subspace C is active if it is in the current context space partition Pt, at time t. For each active context subspace C ∈ Pt, the algorithm maintains the sample mean reward estimates \( rtf(C) \) for each for the predictor for the context arrivals to this subspace. For each active subspace C ∈ Pt, the algorithm also maintains a counter MtC that records the number of context arrivals to C.

In words, the selected predictor has the highest reward estimate for the context subspace Ct among all predictors. This is an intuitive selection based on the sample mean rewards. Next the counter MtC increases by 1 since we have one more sample in C. When the true traffic pattern \( ^*yt \) is revealed (line 6), the sample mean reward estimates for all predictors are then updated (line 7-8). The second part of the algorithm, namely the adaptive context partitioning, is the key of our algorithm (line 9 - 11). At the end of each slot t, the algorithm decides whether to further partition the current subspace Ct, depending on whether we have seen sufficiently many incident arrivals in Ct. More specifically, if \( MtC ≥ A2lp \), then Ct will be further partitioned (line 9), where l is the subspace level of Ct, A > 0 and p > 0 are two design parameters. When partitioning is needed, Ct is uniformly partitioned into 2D smaller hyper cubes (each hypercube is a level-l+1 subspace with side-length half of that of Ct). Then Ct is removed from the active context subspace set P and the new subspaces are added into P (line 11). In this way, P is still a partition whose subspaces are non-overlapping and cover the entire context space. Intuitively, the context space partitioning process can help refine the learning in smaller subspaces. In the next subsection, we will show that by carefully choosing the design parameters A and p, we can achieve sub linear learning regret in time, which implies the optimal time-average prediction performance can be achieved.

**Algorithm 1** Context-aware Traffic Prediction (CATP)
1: Initialize \( P0 = \{ \_ \}, \_rf(\_ ) = 0 \); \( \forall f \in F, M0_0 = 0 \).
2: for each traffic incident (time slot t) do
3: Determine \( Ct \in Pt \) such that \( t \in Ct \).
4: Generate the predictions results for all predictors \( f(\cdot \_t) \); \( \forall f \).
5: Select the final prediction \( yt = f^*(\cdot \_t) \) according to(1).
6: The true traffic pattern \( ^*yt \) is revealed.
7: Update the sample mean reward \( \_rf(Ct) \); \( \forall f \).
8: \( MtC = MtC+1 \).
9: if \( MtC ≥ A2lp \) then
10: \( Ct \) is further partitioned.
11: end if
12: end for

**II. Learning regret analysis**
In this subsection, we analyse the regret of the proposed traffic prediction algorithm. The following technical assumption is needed

**Assumption 1:** For each \( f \in F \), there exists \( L > 0, _- \_ = 0 \) such that for all \( _- \_ \in _- \_ \), we have
\[
|f(\_\_-\_ ) - f(\_\_\_\_ )| ≤ L\|\_-\_\|₂\_ (2)
\]
This states each predictor achieves similar expected rewards (accuracies) for similar contexts. The following theorem establishes the regret bound when the context arrivals are uniformly distributed over the context space, which is the worst case arrival process for regret minimization.

**Theorem 1:** If the context arrival by time T is uniformly distributed over the context space, the regret is upper bounded by
\[
\begin{align*}
D+2_ &
D+3_ 2(D+2_ )l(2LD_ =2 + 2 + \log(T)) + \\
T &
D \\
D+3_ 2D/2K &
∞\Sigma \\
t=0 &\ \ t-2).
\end{align*}
\]
We have shown that the regret upper bound is sub linear in time, implying that the average traffic prediction rewards (e.g. accuracy) achieves the optimal reward as time goes to infinity. Moreover, it also provides performance bounds for any finite time T rather than the asymptotic result. Ensuring a fast convergence rate is important for the algorithm to quickly adapt to the dynamically changing environment.

**Conclusion**
Overall, our findings show that adopting appropriate driving styles can reduce fuel consumption on an average between 3 to 5 litres per 100 km. This can save 15 to 30 litres per bus in just a working day. Taking into account fuel prices, these savings represent 20 to 40 a day, per bus. Considering the working days in a year and with around 1,500 involved drivers, this may impact
significant savings that can go up to 1.5M per year. We proposed a framework for online traffic prediction, which discovers online the contextual specialization of predictors to create a strong hybrid predictor from several weak predictors. The proposed framework matches the real-time traffic situation to the most effective predictor constructed using historical data, thereby self-adapting to the dynamically changing traffic situations. For future work, we plan to extend the current framework to distributed scenarios where traffic data is gathered by distributed entities and thus, coordination among distributed entities are required to achieve a global traffic prediction goal.

References