TRAFFIC FLOW PREDICTIONS EMPLOYING NEURAL NETWORKS IN A NOVEL TRAFFIC FLOW REGIME SEPARATION TECHNIQUE

Mr Stephen Dunne
PhD Researcher
Trinity College Dublin

Dr Bidisha Ghosh
Lecturer
Trinity College Dublin

Abstract
Predictions of fundamental traffic variables in short-term or near-term future are vital for any successful dynamic traffic management application. Univariate short-term traffic flow prediction algorithms are popular in literature. However, to facilitate the operationalities of advanced adaptive traffic management systems, there is a necessity of developing multivariate traffic condition prediction algorithms. A new multivariate short-term traffic flow and speed prediction methodology is proposed in this paper where the traffic flow and speed observations from uncongested (or linear) and congested (or non-linear) regimes are regime-adjusted to ensure consistent system dynamics. The prediction methodology is developed using Artificial Neural Networks (ANN) algorithms in conjunction with adaptive learning rules. These learning rules demonstrate significantly improved accuracy and simultaneous reduction in computation times. Additionally, the paper attempts to identify the most suitable adaptive learning rule from a chosen pool of rules. The validation of the prediction methodology is performed using traffic data from multiple locations in the United Kingdom (UK). The results indicate that the proposed multivariate forecasting algorithm is effective and computationally parsimonious to simultaneously predict traffic flow and speed in freeway or highway networks.

1. INTRODUCTION

Intelligent Transportation Systems (ITS) is a broad spectrum of advanced technologies designed to improve sustainability in existing transportation network by improving efficiency of traffic operationalisation and management. Advanced Traffic Management Systems (ATMS) is an important aspect of ITS and for ATMS to function efficiently, information about the actual and the near-term future traffic state (flow, speed, occupancy etc.) is critical. This necessitates the ability to make and continuously update predictions of traffic flows and link times into the short-term future.

The existing research on short-term traffic variable forecasting can be largely classified either by modelling technique (parametric or non-parametric) or by modelled variables (flow, occupancy, speed etc.). The predominant non-parametric approach studied in the existing literature on traffic variable forecasting is the Artificial Neural Network (ANN) models. Prediction of traffic variables using ANN is a well-researched area and for a comprehensive review of the different approaches and applications [1] can be referred. The conventional ANN structure, such as, the Feed Forward Back Propagation Neural Network (FFBPNN) algorithm has been utilised by researchers to predict traffic flow in near-term future ([2], [3], [4] & [5]). The different structures and applications of ANN in traffic forecasting have established the superiority of these models compared to the other existing methodologies. Due to their ability of precise predictions, adaptability, flexibility and availability of numerous software, ANN models have been utilised in this paper for predicting traffic variables in near-term future.
In existing literature on short-term traffic prediction, the traffic volume has always been a popular choice. Studies on modelling traffic speed are less common ([4] & [6]). However, there is limited research available on short-term traffic speed prediction in comparison with the substantial body of research work available in traffic volume predictions. To address this, a multivariate short-term traffic volume and speed prediction methodology is proposed in this study.

In this paper, a regime-adjusted multivariate dual traffic flow and speed prediction algorithm is proposed. The proposed regime adjustment methodology utilises a time-series classification approach to isolate the observations in congested or non-linear regime and then subsequently preprocesses such information for further prediction. The simplicity of the model enhances its ease of implementation. The effectiveness of the proposed methodology in predicting freeway traffic conditions is explored. The multivariate short-term traffic condition prediction model proposed in this paper utilises the FFBPNN structure in conjunction with adaptive learning rules. In this regard, four different learning algorithms are used and compared in this paper. These include a simple gradient descent (GD) algorithm, a GD algorithm with ALR, a GD algorithm with momentum and an adaptive Levenberg-Marquardt (LM) training algorithm. The FFBPNN structure using adaptive LM training algorithm has been observed to be the most accurate traffic flow predictor in this work. The FFBPNN structure using GD was significantly outperformed, both in terms of error and computational time. This justifies the use of adaptive learning algorithms to train FFBPNN structures in developing traffic condition prediction models.

2. REGIME-ADJUSTED MULTIVARIATE SHORT-TERM TRAFFIC CONDITION FORECASTING

A regime-adjusted multivariate short-term traffic prediction methodology involving a FFBPNN algorithm with adaptive learning strategies has been proposed in this paper to simultaneously predict traffic flow and average traffic speed on highways. To explain the method simply and broadly, initially traffic observations from uncongested (assumed as linear) and congested (assumed as non-linear) regimes are separated and preprocessed. Following this, regime-adjusted flow and average speed are used as input to the FFBPNN structure to obtain forecasts. The prediction methodology is discussed in detail in the following subsections.

2.1 Regime Isolation Methodology

An important feature of the prediction methodology in this work is traffic regime isolation. In this subsection the strategy used for separating the congested (or non-linear) regime from the uncongested (or linear) regime is described. Apart from the points indicating high flow and low speed values, the rest of the observations illustrate a seemingly linear behaviour which can be assumed to be indicative of free-flow conditions (Fig 1). The linearity has also been confirmed quantitatively by fitting a line through the speed flow scatter-plot. The observations which do not follow linearity or more precisely introduces non-linearity in the flow-speed relationship can be identified as the ones belonging to a non-linear or congested regime. These observations can be isolated using a time-series pattern matching approach. The approach is defined and described in the following steps;

Step 1 – Time Series: A time series dataset \( T = t_1, \ldots, t_m \), an ordered set of \( m \) real-valued variables is considered. The data comprising the time-series used in this work is discussed and examined in Section 3.

Step 2 – Subsequence: Given a time series \( T \) of length \( m \), a subsequence \( C_p \) of \( T \) is a sample of length \( w < m \) of contiguous positions from \( T \), that is, \( C_p = t_p, \ldots, t_{p+w-1} \) for \( 1 \leq p \leq m-w+1 \). The process of extracting subsequences from a time series is achieved through the implementation of a sliding window.

Step 3 – Sliding Window: Given a time series \( T \) of length \( m \), and a user-defined subsequence length of \( w \), all possible subsequences can be extracted by sliding a window
of size $w$ across $T$ and extracting each subsequence $C_p$. Using this process, the subsequence following $C_p$ is $C_{p+1} = t_{p+1}, \ldots, t_{p+w}$.

**Step 4 – Pattern Definition:** The pattern of each subsequence $C_p$ is characterised by the slope of the best-fit line generated for each subsequence $C_p$. The change of slope between consecutive subsequences is then measured to identify the similarity of the subsequences.

**Step 5 – Pattern Similarity:** The inclusion of a point in a time-series subsequence which falls outside the linear regime introduces non-linearity which results in a large change of slope in the linear behaviour of the subsequence and also provides a low $R^2$ value for the best-fit line, indicating non-linear behaviour. Using this assumption, the points introducing non-linearity in the traffic time-series data are identified.

**Step 6 – Time Series Classification:** A subsequence $C_p$ which is characterised by a $R^2$ value of a beyond a defined lower limit and the difference of slope, $\theta$, between $C_p$ and $C_{p-1}$ is beyond a threshold $T$ classifies the sequence as non-linear. The threshold $T$ can be the lower limit of the 95% confidence interval of a change of slope distribution calculated using historical time-series datasets. The lower limit of the $R^2$ value, $L$, can be the average value calculated using historical time-series datasets.

The abovementioned steps are followed to identify the traffic observations located in the congested or non-linear regime. The observations in the congested regime are preprocessed to introduce an adjusted surrogate linear speed-flow calibration to ensure consistent traffic dynamics. It can be seen, that each set of information on traffic volume, average speed and time of observation, $t$, is a vector that can be represented as $X(q,u,t)$.

The observations belonging to the congested regime are adjusted using the following conditions,

$$
\text{for } \theta < T \text{ and } R^2 < L, \\
X(q,u,t_i) = E(X(q,u,t_i)), \quad i = 1,\ldots, MN
$$

where, $X(q,u,t_i)$ defines the traffic time-series at flow $q$, speed $u$ and time $t$ of a day; $MN$ is the number of days at which traffic data is modelled, $E$ denotes the expectation. The observations in the uncongested regime are not preprocessed.

### 2.2 Artificial Neural Networks (ANN)

Mathematically, the neuron $j$ can be described as follows,

$$
u_q = \sum_{p=1}^{N} w_{pq} x_p \quad a_q = \varphi(u_q + b_q)
$$

where $x_1, x_2, \ldots, x_p$ are the input signals, $w_{pq}$ is the connection weight from neuron $p$ in layer $l$ to neuron $q$ in layer $l+1$, $u_q$ is the linear combiner output due to the input signals, $b_q$ is the bias, $\varphi(\cdot)$ is the activation function and $a_q$ is the output signal of the neuron [7].

#### 2.2.1 Feed Forward Back Propagation Neural Network (FFBPNN)

The FFBPNN model consists of both a forward and backward phase. The input vector contains $N$ elements which is equal to the number of time instants at which traffic data is collected within a day. Output values are calculated using equation 2 and a log-sigmoid function is used as the activation function in this case. The Back Propagation (BP) phase of the network compares the network outputs $a_{pq}$, calculated with the target values $t_{pq}$, and is an iterative optimization of the error function that represents the performance of the network. This function of error, $E$, is defined as
where $N$ and $MN$ are defined as before. $E$ is minimised using the gradient descent optimisation technique. The partial derivative of the error function in relation to each weight provides a direction of steepest descent. The corrections to the connection weights within the network are determined for each iteration using this partial derivative. The weight updation equation is,

$$w_{pq}(k+1) - w_{pq}(k) = \Delta w_{pq}(k) = -\eta \frac{\delta E(k)}{\delta w_{ipq}(k)}$$ (4)

where $\eta$ is the learning rate, a small positive constant. If the difference between network output and desired output becomes negligible or acceptable then the learning process terminates.

2.2.2 Adaptive Learning Rate Model (ALRM)

The introduction of an ALR, as shown in equation 5, allows the learning rate to be adjusted during the running of the BP algorithm.

$$\eta(k+1) = \eta(k) \times \eta_{\text{factor}} \quad \eta_{\text{factor}} = \begin{cases} 1.05, & \Delta E_a > 0 \\ 0.7, & \text{otherwise} \end{cases}$$ (5)

where $\eta_{\text{factor}}$ is an adaptive multiplicative factor which alters the value of $\eta$ depending on how the error value has changed in the previous iteration. In this study, two values of $\eta_{\text{factor}}$ are contained within the ALR algorithm; one which increases the value of $\eta$ if the error has decreased over the previous iteration and vice versa (Equation 5).

2.2.3 Levenberg-Marquardt Model (ALRLM)

The second learning algorithm investigated in this paper is the Levenberg-Marquardt (LM) algorithm ([8] & [9]). This algorithm is an iterative optimisation technique that locates the minimum of a multivariate non-linear function. The error function in equation 3 can be rewritten as a function of the weights of the network as,

$$f(W) = \frac{1}{2} E^T E = \frac{1}{2} E^T (W) E^T (W) = \frac{1}{2} \sum_{p=1}^{MN} \sum_{q=1}^{N} (t_{pq} - a_{pq})^2,$$ where $W \equiv [w_1, w_2, w_3, \ldots, w_p]^T$ (6)

and $E = [e_{11}, \ldots, e_{MN}]$. $W$ consists of all the weights of the network. The weight updation is then achieved as,

$$W(k+1) = W(k) - (J^T J + \lambda I)^{-1} J^T E_k$$ (7)

where the Jacobian matrix $J_k$ contains the first derivatives of the network errors with respect to weights and biases, and $E_k$ is a vector of network errors. ($J^T J$) is positive definite, but if it is not, then some perturbations made into it control the probability of it being non positive. Thus,

$$W(k+1) = W(k) - (J^T J_k + \lambda I)^{-1} J^T_k E_k$$ (8)

The quantity $\lambda$ is a learning parameter which decreases as the iterative process approaches to a minimum. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one (i.e. large $\lambda$), the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton (GN).
method ($\lambda = 0$). It is this value $\lambda$ that displays adaptive abilities. This LM algorithm is referred to as ALRLM in the paper.

### 2.2.4 Adaptive Learning Rate with Momentum Model (ALRMM)

The introduction of momentum (Equation 9) alters the weight alteration as follows:

$$\Delta w_{pq}(k) = -\eta \frac{\delta E(k)}{\delta w_{pq}(k)} + \alpha \Delta w_{pq}(k-1)$$

(9)

where $\alpha$ is the momentum factor, 0.9 in this work. Momentum alters the connection weight updating procedure. The momentum factor determines the influence that the previous weight change has on the current weight updation. Momentum produces an averaging effect whereby changes in connection weights take into account both the current error value as well as previous weight changes, leading to a more considered approach to weight updation.

### 3. APPLICATION OF THE PROPOSED PREDICTION ALGORITHM

#### 3.1 Traffic Data

The data modelled using the proposed prediction methodology is obtained from the MIDAS database of the UK Highways Agency. MIDAS datasets contain traffic volume, average speed, headway and occupancy observations aggregated over every 60 seconds in every lane of a given motorway section. For this paper, 10 days of traffic data from January 2010 are collected from two motorway sites, titled “A1/9340A” and “M6/6615A”. There are two lanes in each direction in the motorway sections chosen. In both cases, traffic observations from the northbound lanes of the chosen motorway sections are modelled.

![Flow-Speed Scatterplots for the four motorway sites](image_url)

**Figure 1:** Flow-Speed Scatterplots for the four motorway sites

It is important to note that the traffic flow observations recorded on a weekday are substantially different to those traffic observations recorded on the weekend. Therefore for
consistency purposes, only weekday traffic condition observations are modelled in this paper. The values of average speed stay within narrower confines than that of the traffic flow. However, large variations of average speed over small time periods do occur, notably in Lane 2 of the M6/6615A motorway section. Traffic flow peaks at different times at different motorway locations. Models have been fitted for five different time aggregations; 1 minute, 5 minute, 15 minute, 30 minute and 1 hour. The prediction methodology described in section 2 is heavily dependent on the flow-speed relationship among the bi-variate traffic time-series data (Fig. 1).

3.2 Model Fitting

The model fitting was performed in two stages. At the first stage the bi-variate traffic time-series datasets from 4 sites were analysed to identify (i) Length of a subsequence, (ii) Change of slope threshold and (iii) $R^2$ value thresholds. On identification of the thresholds, the points identified within the non-linear regime were preprocessed using equation 1. Further to preprocessing, the traffic data was used as input to the ANN structures described in section 2.2 for prediction. At the second stage 20 FFBPNN structures in total had been used to predict traffic volume and average speed simultaneously in a multivariate paradigm. The 20 models comprise 3 different adaptive learning algorithms and the original GD learning rule used to predict traffic flow and average speed at 5 different time aggregation intervals. The number of elements in the input vector, $N$, corresponds the time aggregation level of the data. For all networks the output vector is same size as the input vector. The training datasets are composed of regime-adjusted traffic flows and average speeds from 10 consecutive weekdays for all 20 models. Traffic observations from the first 7 days are used to make up the training input. Observations from day 2 to day 8 are used as the training target. Once the network is trained on this training dataset, a new day (Day 9), previously unseen by the networks, is presented as the test input. The simulation of the network with this test input results in the prediction of traffic flow and average speed information as the test output. This is then compared with the test target (Day 10), and error values are generated based on the difference between output and target.

4. FORECASTS

The proposed prediction methodology is used to predict traffic flow and speed data at the four chosen locations as described in the previous section. The prediction accuracy of the proposed model in the form of MAPE (Mean Absolute Percentage Error) values is presented for v-step ahead predictions (Table 1).

<table>
<thead>
<tr>
<th></th>
<th>Average Speed</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GDM</td>
<td>ALRM</td>
</tr>
<tr>
<td>1 Minute</td>
<td>MAPE</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>53.79</td>
<td>18.44</td>
</tr>
<tr>
<td>5 Minute</td>
<td>MAPE</td>
<td>Time</td>
</tr>
<tr>
<td>15 Minute</td>
<td>MAPE</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>18.78</td>
<td>9.74</td>
</tr>
<tr>
<td>30 Minute</td>
<td>MAPE</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>16.09</td>
<td>15.83</td>
</tr>
<tr>
<td>1 Hour</td>
<td>MAPE</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>14.32</td>
<td>8.93</td>
</tr>
</tbody>
</table>

The 20 different ANN models with 4 different learning strategies were compared to identify which of the models produce the most accurate forecasts, and also whether an increase in prediction accuracy compromised the computational speed of the model. The MAPE values and computation time for A1/9340A Lane 1 are shown in Table 1. From both flow and average speed prediction errors, it can be ascertained that at the majority of time-
aggregations, the GDM algorithm produces the least accurate forecasts as well as the longest computation times. The ALRM, ALRMM and ALRLM all produce competitive forecasts and computation times. Overall the computation times for all algorithms decrease with increasing time aggregation levels, as the size of the input vector reduces accordingly. It can be concluded that an increase in prediction accuracy does not compromise the computation speed from an algorithm perspective, rather the MAPE and computation times all follow the same trend. For all 4 motorway sites, the MAPE values correlate with the COV values. In general, the average speed MAPE values are less than the corresponding flow MAPE values in every instance. In a motorway situation, the average speed varies over a comparatively smaller range than the traffic flow as can be seen in Fig. 1 and quantitatively it is evident from much lower COV values for speed than for flow throughout the datasets. However, a direct correlation between MAPE, and COV values should not be assumed. COV can only be considered as an approximate indicator of prediction accuracy.

For illustrative purposes, the prediction results using ALRLM at A1/9340A Lane 1 for 15 minute data interval have been shown in Fig. 2.

Figure 2: (a) Flow Prediction, (b) Speed Prediction

The traffic flow and speed forecasts along with the original observations are shown in Fig. 2(a) and Fig. 2(b) respectively. The proposed regime-adjusted ANN based prediction methodology provides reasonably accurate forecasts for the bivariate traffic time series. The same traffic flow data when modelled using an ordinary univariate FFBPNN (without regime adjustment and no ALR element) results in MAPE values equal to 20%. This proves that the proposed model provides better prediction accuracy through regime adjustment and learning rate improvement. The MAPE values for flow and speed predictions from regime-adjusted ALRLM algorithm are plotted against the time intervals in Fig. 3 for all 4 sites.

Figure 3: MAPE vs. Time Aggregation (a) Flow, (b) Speed
In all cases, the sharpest reduction in MAPE occurs between the finer (1 min, 5 min, 15 min) time aggregations. The change in MAPE values for time aggregations greater than 15 minutes is negligible compared to the finer aggregations. The curves for all 8 cases (flow and speed for all 4 sites) behave in the same fashion. At higher resolutions, there is nearly an exponential decrease with increased aggregation, and at lower resolutions, the curve behaves asymptotically. This establishes that beyond a certain level, time aggregation does not influence the prediction levels greatly. The curves suggest that 15 minutes is the optimum forecasting time aggregation. However, more detailed analysis into time aggregation levels and signal-to-noise ratios in traffic time-series is necessary to arrive at a definitive optimum time aggregation level.

5. CONCLUSION

In this paper, a regime based multivariate traffic flow and speed forecasting algorithm is proposed. This study is one of the first instances of application of adaptive learning strategies such as ALR or momentum in ANN based multiple traffic condition forecasting algorithms. The algorithm uses an ANN structure with three adaptive learning strategies as well as the conventional GD learning rule. The forecasts, the precision errors and the computation times of the ANN structure with four different learning rules were compared to establish the most efficient prediction strategy. The ANN structure using the adaptive LM learning algorithm proves to be the best prediction methodology to predict traffic flow and average speed simultaneously. The proposed algorithm is capable of forecasting traffic data a day ahead in future.

The main contribution of the proposed methodology lies in providing a regime adjustment methodology to predict traffic data. This algorithm isolates the traffic data in the congested or nonlinear regime from the ones in the uncongested or linear regime and the isolated observations are preprocessed to achieve stable predictions through the above mentioned ANN structure. The proposed algorithm has been evaluated for modelling traffic observations on highways. As a future work, the prediction methodology can be applied to urban arterials with more significant congestion levels and also for the effect of traffic signals.

The study also identifies the effect of time-aggregation on MAPE values when predicting short-term traffic condition related variables. Traffic flow prediction improves with the increase in aggregation time interval as the variability of the data reduces considerably with lower resolution. The same behaviour is exhibited in the average speed predictions, with the underlying trend being that as time-aggregation increases, the accuracy of prediction increases too. It is important to note that, the prediction error do not have a linear relationship with the time aggregation levels. This relationship is generally site and flow dynamics specific.

REFERENCES