SIMULATION OPTIMIZATION OF CAR-FOLLOWING MODELS USING FLEXIBLE MODELS

Vasileia Papathanasopoulou¹, Constantinos Antoniou²∗

¹National Technical University of Athens
Laboratory of Transportation Engineering
GR15780, Zografou, Greece
vasileia.papathanasopoulou@gmail.com

²National Technical University of Athens
Laboratory of Transportation Engineering
GR15780, Zografou, Greece
antoniou@central.ntua.gr

Keywords: Car-following Models, Data-driven Approaches, Classification, Clustering, Gipps’ model, Locally Weighted Regression (Loess), Machine Learning Methods, Speed Estimation, Intelligent Transportation Systems

Abstract. Car following behavior is a key component of microscopic traffic simulation. Numerous models based on traffic flow theory have been developed for decades in order to represent the longitudinal interactions between vehicles as realistically as possible. Nowadays, there is a shift from conventional models to data-driven approaches. Data-driven methods are more flexible and allow the incorporation of additional information to estimation of car-following models. On the other hand, conventional car-following models are founded on traffic flow theory, thus providing better insight into traffic behavior. The integration of data-driven methods in applications of intelligent transportation systems is an attractive perspective. Towards this direction, in this research an existing data-driven approach is further validated using another training dataset. Then, the methodology is modified, extended and enriched so that an improved methodological framework to be suggested for the optimization of car-following models. Machine learning techniques, such as classification, locally weighted regression (loess) and clustering, are innovatively integrated. In this paper, validation of the proposed methods is demonstrated on data from two sources: (i) data collected from a sequence of instrumented vehicles in Naples, Italy, and (ii) data from the NGSIM project. In addition, a conventional car-following model, the Gipps’ model, is used as reference in order to monitor and evaluate the effectiveness of the proposed method. Based on the encouraging results, it is suggested that machine learning methods should be further investigated as they could ensure reliability and improvement in data driven estimation of car-following models.
1 INTRODUCTION

Simulation models play an important role in traffic engineering and recently in the development of Intelligent Transportation Systems (ITS) [1]. They are divided into microscopic, mesoscopic and macroscopic according to the detail of traffic flow that is used in the modeling. Microscopic models describe in high level of detail interactions between individual vehicles, including interactions between vehicles and roads as well [2]. They consist of lane changing, gap-acceptance, overtaking, speed adaptation, ramp merging and car-following models [3]. On the other hand, macroscopic models represent traffic states in a lower level of detail using aggregated variables (traffic flow, density, speed) and theories of fluid dynamics [4]. Finally, mesoscopic models provide an intermediate level of detail using speed-density relationships and queuing models [4].

Appropriate models are chosen according to the requirements of the application that will be used in. This research is directed at microscopic traffic simulation which gives the opportunity of detailed analysis required in the development of some complex applications and Intelligent Transportation Systems. Focusing on optimization of car-following models, the key elements of microscopic simulation [1,5,6], an alternative methodological framework is suggested. Car-following models generally represent driving behavior influenced by the preceding vehicle moving in the same lane so as a crash to be avoided. According to Olstam and Tapani [3], they are grouped into categories such as Gazis-Herman-Rothery models [7], safe distance models [8,9], psycho-physical models [10,11] and fuzzy logic models [12,13].

Over the years, many researches have been demonstrated aiming at the optimization of car-following models. Recently, it has been clarified that driving behavior varies in different traffic conditions, such as free-flowing, approaching, emergency braking, and stop-and-go, etc. [1,18-21]. Therefore, there has been a shift from single state models [14,15] to more flexible models. The lack of models capable of capturing various traffic states and correspondingly various driving behaviors has led to the development of multi-regime approaches [16]. Nowadays, the generalization of these multi-regime approaches is a challenge issue.

Restrictions, related to the number of regimes and their complexity, have been the motivation for this research on estimation of car-following models. An alternative methodology based on data-driven approaches is proposed; actually an existing methodology has been modified to address these problems. Data-driven methods have been already used in applications in the field of transportation [e.g. 22-28]. These methods are more flexible than conventional models and allow the incorporation of additional information. The development of data-driven methods has also been benefited from technological advancements such as differential GPS and real time kinematic which allow the collection and the availability of high quality traffic data [29] (more examples could be found in [30]).

In this paper an existing methodology based on a machine learning method is further validated and enriched for optimization of car-following models. The historical background of car following models and the development of data-driven approaches is first presented. The existing methodology is applied to a number of available NGSIM data sets and the different nature of data is discussed regarding the impact on the efficiency of the method. In addition, the existing methodology is further extended and improved for the development of more reliable car following models. The revised innovative methodology integrates data-driven methods such as classification, loess method and clustering, and is validated to Naples data. Finally, the results are discussed and conclusions are drawn.
2 HISTORICAL BACKGROUND

A historical review of car following models has been performed by Brackstone and McDonald [5]. Reuscher [14] and Pipes [15] introduced the idea of car-following models. Representative microscopic traffic models between the 1950s and the 1970s have been developed by [8,17,31-38]. Most of them are defined by an acceleration function, which includes the difference of position \( x_{i+1} - x_i \) and the difference of speed \( v_{i+1} - v_i \) between a vehicle \( i \) and its lead vehicle \( i+1 \): the difference of position \( x_{i+1} - x_i \) and the difference of speed \( v_{i+1} - v_i \). Other models have been developed including only one variable such as the difference of speed [17,31] or the difference of position [35]. Gazis et al. [7] proposed a General Motors model (GM) with doubtful efficiency both in low and high-speed networks [16]. Several extensions to the GM framework followed [39]. Wiedemann [40] and Leutzbach [41] introduced psycho-physical models in order to address restrictions of GM models. Wiedemann and Reiter [10] suggested that there are longitudinal interactions in four traffic states: free flowing, approaching, car following and emergency situation.

After 1990, Tordeux et al. [42] identified a different tendency for development of car following models due to technological advancements and computer simulation. New microscopic methods are considered as multi-agent and are defined by a system of differential equations, each of which captures a different state. Treiber et al. [43] clarified that reaction time and time steps should have various values in the simulation process. Gipps’ model [9, 44] is a safety distance model described by two speed equations correspondingly to two traffic states, the free flowing and car following state. In this research this model is used as a reference for the framework developed in this research. Rakha and Wang [45] tried to modify Gipps’ model. A detailed analysis of the model evolution is presented by Ciuffo et al. [46], Bando et al. [47,48] developed a nonlinear model, the Optimal Velocity model, to deal with stop-and-go traffic states. Further research was performed later by [49-55].

According to Subramanian [56,1], drivers’ reaction time is differentiated under acceleration or deceleration conditions. Ahmed [19] suggested an acceleration model both for free-flowing and car-following situations. Newell [57] clarified that the trajectory of a vehicle depends on a time and a minimum distance of spacing. Treiber et al. [58] proposed the Intelligent Driver Model, which determines driver’s acceleration in relation with the gap, the speed and the speed difference between a pair of vehicles moving in sequence. Aw et al. [59] proposed a new general model. Zhang and Kim [60] developed a multi-regime car-following model, which is determined by a gap-distance function and the traffic state. Hamdar and Mahmassani [61] demonstrated calibration and validation of existing car-following models using NGSIM data. Tordeux et al. [42] proposed the impact of the vehicle type on driving behavior. Moreover, the assumption of the GM model that a driver will accelerate if the speed of the preceding vehicle is higher is re-examined.

According to researches, more and more parameters and traffic states should be integrated in simulation process. This need has led to the development of multi regime models and by extension to data-driven approaches. A multimodal regression to speed-flow data has been performed by Einbeck and Tutz [62]. Sun and Zhou [63] used cluster analysis in order to determine the regime boundaries for traditional speed–density models. Antoniou and Koutsopoulos [22] suggested a data-driven approach as an alternative to the classic speed–density models. Zhang et al. [31] have demonstrated the use of machine learning methods to support the development of data-driven intelligent transportation system. Data-driven approaches have already been used in a fully adaptive cruise control system by Bifulco et al. [64] or in car-following modeling with artificial neural networks by Colombaroni and Fusco [65]. Finally, Papathanasopoulou and Antoniou [66] have performed a data-driven approach based on
loess method for speed estimation using Naples data. This research is further extended in this paper.

3 METHODOLOGY

3.1 Methodological Framework

Two data-driven approaches are presented in this paper, outlined in Figure 1. Regarding the first one approach is an existing method based on locally weighted regression (loess) which has been already proposed and analyzed in an earlier research [66]. The second data-driven method is an extension and improvement of the earlier method and comprises a combination of computational methods, such as locally weighted regression, model-based clustering and classification.

Both methodological approaches include two parts: training and application. In the training step the estimation of car-following models is achieved using a training dataset with triples $<v_i, v_{i-1}, d_{i,i-1}>$ (leader and follower speed and their distance) per each time instant. The problem to be addressed is the speed estimation of the third vehicle when there are available the speeds of the preceding and the following vehicle and its distance from them. In the application process, when new observations arise, the appropriate calibrated models are retrieved from the knowledge base and are applied to provide speed predictions $v_i$ for the following vehicle and the next time instant. The proposed methods rely on non-parametric approaches and do not include any fixed functional form. They might be considered as generalization of the multi-regime approaches [22,23].

As concerns as the second methodological approach, it includes a clustering step to identify portions of the available data that correspond to traffic states with similar characteristics. Then, a locally weighted regression is applied to each cluster separately and representative models are formed for each group fitting to the data (fitting). The application step follows,
when new measurements arise. New data are classified to the appropriate classes based on their characteristics. The flexible model that has been estimated for that class is then retrieved from the knowledge base and applied to the new data for the estimation of the speeds of the following vehicle.

The performance of each approach is evaluated using the root-mean square error (RMSN) of speeds. This assesses the overall error of each method estimating the difference between the observed \(Y_{o}^{obr}\) and simulated values \(Y_{o}^{sim}\), \(N\) is the number of observations [67,68]:

\[
RMSN = \sqrt{\frac{\sum_{n=1}^{N} (Y_{o}^{obr} - Y_{o}^{sim})^2}{\sum_{n=1}^{N} Y_{o}^{obr}}},
\]

(1)

### 3.2 Methodological components

**Locally weighted regression** could be considered as a generalization of the k-nearest neighbor method [69]. It was firstly introduced by Cleveland [70] and the following analysis is based on [71]. Locally weighted regression \(y_{i}=g(x_{i})+\varepsilon_{i}\), where \(i=1,\ldots,n\) index of observations, \(g\) is the regression function and \(\varepsilon_{i}\) are residual errors, provides an estimate \(g(x)\) of each regression surface at any value \(x\) in the \(d\)-dimensional space of the independent variables. Correlations between observations of the response variable \(y_{i}\) and the vector with the observations \(d\)-tuples \(x_{i}\) of \(d\) predictor variables are identified. Local regression provides an estimation of function \(g(x)\) near \(x = x_{0}\) according to its value in a particular parametric class. This estimation could be achieved by adapting a regression surface to the data points within a neighborhood of the point \(x_{0}\), which is bounded by a smoothing parameter: span \(\alpha\). The span determines the percentage of data that are considered for each local fit and hence the smoothness of the estimated surface is influenced [72]. Each local regression uses either a first or a second degree polynomial that is specified by the value of the “degree” parameter of the method.

The data are weighted according to their distance from the center of neighborhood \(x\), therefore a distance and a weight function are required. As a distance function \(p\), Euclidean distance could be used for a single independent variable; otherwise, for the multiple regression case, any variable should be evaluated on a scale before applying a standard distance function [73].

A weight function defines the size of influence on fit for each data point taking for granted that nearby points have higher influence than the most distant. Therefore the weight function calculates the distances between each point and the estimation point and higher values in a scale from 0 to 1 are set for the nearest observations. A weight function should meet the requirements determined by Cleveland [70] and the most common one is the tri-cube function:

\[
W(u) = \begin{cases} 
(1-u)^3, & 0 \leq u \leq 1 \\
0, & \text{otherwise} 
\end{cases}
\]

(2)

The weight of each observation \((y_{i}, x_{i})\) is defined as following:

\[
w_{i}(x) = W[p(x, x_{i}) / d(x)] = (1 - \frac{(x_{i} - x)^{3}}{d(x)})
\]

(3)

where \(d(x)\) is the distance of the most distant predictor value within the area of influence.
In the loess method, weighted least squares are used so as linear or quadratic functions of the independent variables could be fitted at the centers of neighborhoods [70]. The objective function that should be minimized is:

$$\sum_{i=1}^{n} w_i \cdot e_i^2$$  \(\text{(4)}\)

One of the most common methods of classification is k-nearest neighbors [69]. According to this method, all observations correspond to points in n-dimensional space. Future data points are registered in the class of nearest neighbors of the already grouped data. Especially, the point of the nearest neighbor classification is the calculation of the correlation map:

$$f(z) = \arg \min_{y \in M} d(z, y)$$  \(\text{(5)}\)

In a pattern space \(P\), where \(M \subseteq P\), \(z \in P\) and \(d()\) is a metric in P-dimensional space.

The evaluation of Eq. (5) could be easily achieved on a computer following three steps: computation of an array with distances from \(z\) to each \(y \in M\), finding the minimum distance after comparisons and exporting the final result \(y^*\) [74].

The nearest neighbors could be defined according to the Euclidean distance [75], if a point \(x\) is described as \(<a_1(x), a_2(x), \ldots, a_n(x)>\) where \(a_r(x)\) corresponds to the value of the \(r\)-th attribute of \(x\). Attributes of \(x\) could include density, traffic flow, and time. The distance between two points is defined as follows [69]:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} [a_r(x_i) - a_r(x_j)]^2}$$  \(\text{(6)}\)

Thus the class of a new observation \(x_i\) is the same as the class of point \(x_j\), which minimizes the distance \(\|x_i-x_j\|\) [75].

Fraley and Raftery [76,77] suggest a model based clustering which combines hierarchical clustering, expectation-maximization algorithm (EM algorithm) for mixture models and Bayesian information Criterion (BIC) for selection of models and number of classes [78]. Hierarchical clustering, used for model-based hierarchical agglomeration, is initialized by default with each observation of the data in a cluster by itself and finished when all observations have been merged into a cluster. A classification maximum likelihood approach is required to determine which two groups are merged at each stage [79-83].

EM algorithm is included in the R Mclust package and is applied for maximum likelihood clustering with parameterized Gaussian mixture models [82,83]. The EM algorithm is implemented in two steps: E-step which calculates a matrix \(z_{ik}\), which corresponds to the likelihood of an observation \(i\) to be merged into a cluster \(k\) given the current parameter estimates, and M-step, which calculates maximum likelihood parameter estimates given \(z\). Each cluster is represented by a Gaussian model \(\phi_k(x|\mu_k, \Sigma_k)\), where \(x\) are the data, \(k\) an integer indicating a cluster centered at means \(\mu_k\) and covariances \(\Sigma_k\). Then the maximum likelihood values for the Gaussian mixture model will be [76]:

$$\prod_{i=1}^{n} \sum_{k=1}^{G} \tau_k \phi_k(x_i \mid \mu_k, \Sigma_k)$$  \(\text{(7)}\)

where \(\tau_k\) are the mixing proportions.

Banfield and Raftery [79] suggested a clustering strategy based on a maximization algorithm and Bayes factors. This strategy was upgraded by [76,77,81] and could be carried out with the following steps:
• A maximum number of clusters and a subset of covariance structures are considered
• A hierarchical agglomeration that maximizes the classification likelihood for each model is performed and the appropriate classifications are illustrated up to M groups.
• The EM algorithm is applied for each model and each number of clusters 2,…, M. The procedure is initialized from the classification result of hierarchical agglomeration.
• The Bayesian information Criterion BIC is calculated for the one-cluster case for each model and for the mixture model with the optimal parameters from EM for 2,…, M clusters. Each combination corresponds to a unique probability model.
• The model with the highest BIC is selected and the best classification is recovered. Although in such a way the optimal number of classes is determined, a lower number of classes could be chosen, aiming at the development of more parsimonious models.

4 EXPERIMENTAL SET-UP

The data used in this survey are available from two sources: (i) an experiment carried out in Naples, Italy [84] and (ii) from the «Next Generation SIMulation (NGSIM) program» [85]. Naples Data are used for the validation of the second methodological approach, while NGSIM data for further validation of the first methodological approach.

4.1 Naples Data

A series of data-collection experiments were carried out on roads surrounding the city of Naples, in Italy [84]. All data were collected under real traffic conditions in October 2002. Although traffic conditions and driving routes may be different in each dataset, the platoon consisted of four vehicles is unchanged regarding the vehicles, the drivers and the sequence. Datasets with index A, C correspond to one-lane urban road, while datasets with index B to a two-lane extrarural highway. However, all selected roads have one lane per direction in order to avoid effects on driving behavior by lane changing. GPS receivers located on the vehicles were recording the coordinates X, Y, Z of each vehicle per 0.1s (in 10Hz). Thus, the speed of each vehicle and the distances between each pair of vehicles could be calculated at each moment. The setup included five dual frequency GPS+GLONASS receivers (1 base station + 4 rovers) with expected accuracy in real time kinematic 10mm+1.0 ppm horizontally and 15 mm+1.0 ppm vertically.

In this research, data used are readily available observations from the field. No corrections and no interpolation have been occurred. Therefore, only segments with consecutive measurements have been taken into consideration. Six data series were used, one for calibration and five for validation. A detailed description of the data could be found in Punzo et al. [84], who kindly provided the data for this research.

4.2 NGSIM Data

The “Next Generation SIMulation (NGSIM)” program (http://ngsim.fhwa.dot.gov) includes vehicle trajectories in real traffic conditions, which –along with other output of the project- have become available to the scientific community for research of microscopic driving behavior. As this data-set is rather different than the Naples data (different road type, vehicle fleet composition and driving population) it provides an opportunity to assess the transferability of the car-following models that were estimated on the Naples data.

The considered NGSIM data were collected on eastbound I-80 in the San Francisco Bay area in Emeryville on April 13, 2005 [83]. The study area extends approximately 500m in length and consists of six freeway lanes. Seven modern digital cameras were mounted on the top of a 30-story-building adjacent to the freeway and were recording passing vehicles. The
custom NG-VIDEO software application transformed video to vehicle trajectories data (also at 10Hz). These data were recorded mainly in congested conditions. 45 minutes of data are available in a data set divided into three periods of 15 minutes and particularly in accordance with the register time, 4:00 p.m. to 4:15 p.m., 5:00 p.m. to 5:15 p.m., and 5:15 p.m. to 5:30 p.m.

For each vehicle the available data which are taken into account are: vehicle ID, type of vehicle (only cars are taken into consideration), time (ms), global coordinate X (feet), global coordinate Y (feet), length of vehicle, vehicle velocity (feet/s), distance between the front side of a vehicle and the front side of the preceding vehicle, number of the preceding vehicle, number of the following vehicle, lane identification. (The data were converted to SI units prior to our application). Due to the large amount of available NGSIM data, 17 tetrads of vehicles moving consecutively were selected randomly for this analysis. The vehicles, which compose a tetrad, are considered only when they are moving in the same lane and in sequence one after the other. This is easily recognizable from the lane identification and the number of preceding and following vehicles.

NGSIM data have been used in many studies for calibration or validation of existing models or algorithms (e.g. [86]). It is also noteworthy that in the years 2007-2008 more than 30 studies used the NGSIM data [87]. However, only few studies have raised the issue of their accuracy [58,87,88]. Although the way that the velocities and accelerations of vehicles were calculated and the errors were reduced is not known, studies suggest the existence of residual noise and errors in the data [86,87]. In the context of this work the existence of noise in data is not addressed, presuming that if there are errors, they are included in both methods (model Gipps, proposed method) and therefore may not affect the comparison but the result of each method separately. Also, this implies that the presented approach can work directly with collected data, without requiring copious data-cleaning efforts.

5 VALIDATION RESULTS

The first methodological approach has been already demonstrated using Naples data by [66]. The authors have presented a sensitivity analysis both of Gipps’ model and Loess method and their calibration process as well. For Gipps’ model the following two combinations of parameters have been chosen as optimal: \( \tau=0.4 \) s, \( V_n=14 \) m/s, \( \alpha_n=0.8 \) m/s\(^2\), \( s_{n-1}=5.6 \) m, \( b_n=-5.2 \) m/s\(^2\) and \( b=-3.0 \) m/s\(^2\) or for a more reasonable value of time reaction \( \tau=1.0 \) s, \( V_n=16 \) m/s, \( \alpha_n=1.6 \) m/s\(^2\), \( s_{n-1}=5.6 \) m, \( b_n=-5.2 \) m/s\(^2\) and \( b=-3.0 \) m/s\(^2\). For Loess method degree=1 and span=0.75 have been specified.

<table>
<thead>
<tr>
<th>Data series</th>
<th>Reaction time ( \tau=0.4s )</th>
<th></th>
<th>Reaction time ( \tau=1.0s )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSN (%)</td>
<td>Improvement of estimation (%)</td>
<td>RMSN (%)</td>
<td>Improvement of estimation (%)</td>
</tr>
<tr>
<td>B1695</td>
<td>Gipps’ models</td>
<td>Loess method</td>
<td>Gipps’ models</td>
<td>Loess method</td>
</tr>
<tr>
<td></td>
<td>2.7</td>
<td>1.6</td>
<td>40.7</td>
<td>4.9</td>
</tr>
<tr>
<td>C621</td>
<td>6.6</td>
<td>4.3</td>
<td>34.8</td>
<td>14.4</td>
</tr>
<tr>
<td>A358</td>
<td>2.7</td>
<td>2.1</td>
<td>22.2</td>
<td>12.7</td>
</tr>
<tr>
<td>A172</td>
<td>4.6</td>
<td>3.4</td>
<td>26.1</td>
<td>16.0</td>
</tr>
<tr>
<td>C168</td>
<td>2.3</td>
<td>1.8</td>
<td>21.7</td>
<td>4.9</td>
</tr>
<tr>
<td>C171</td>
<td>7.2</td>
<td>6.2</td>
<td>13.9</td>
<td>31.6</td>
</tr>
</tbody>
</table>

Table 1: Results for speed estimation for all Naples data sets using the first methodological approach.

Both methods have been calibrated using the most representative data series B1695 and for the speed estimation the same factors have been used (speed \( v_2(t) \) and \( v_3(t) \) of vehicles 2 and
3 and distance $D_{23}(t)$ between vehicles 2 and 3). The results encourage the application of the data-driven approaches and are summarized in Table 1. Loess method outperforms Gipps’ model for all the available data series.

The calibrated models are now validated to another data set from the US (NGSIM data) and it is demonstrated how the different nature of data affect the accuracy of speed estimation. NGSIM data and data from Naples are of different nature, as the former refer to freeway in congestion conditions and the latter to roads with one lane per direction. Moreover, as Marsden et al. [89] suggested, differences between car-following headways and times-to-collision are identified between different sites. In this section, the transferability of the models estimated in Naples to the NGSIM data set is tested. Two models are presented: (i) Gipps’, (ii) a loess model with the same data as those used by Gipps’ model.

The results are presented in Figures 2, 3 for time reaction 0.4 s and 0.1 s accordingly and several observations can be drawn. As expected, the RMSN values are higher than in the Naples data, as model calibration and validation/application was performed on dissimilar data. The proposed loess method seems to provide better results than Gipps’ model. The machine learning approach seems to be more robust, while the effectiveness of the conventional car-following models may depend significantly on the chosen parameter values. Using additional data would be easy with the proposed data-driven model and improves the performance even further; on the other hand, reformulating Gipps’ model to consider additional parameters would be a tedious exercise.

![Figure 2: Comparison of RMSN by applying Gipps’ model and loess method for NGSIM data for reaction time $\tau=0.4$ s](image-url)
The degree by which the proposed approach outperforms the reference model varies across data series. In order to develop some insight into this, an exploration of the speed profiles of the various vehicles was performed.

Figure 4 presents the speed profile for the considered vehicle in the longest sequence of the Naples data-set (B1695) used for calibration, while Figures 5 and 6 present similar speed profiles for data series that showed satisfactory performance (Fig. 5) and less satisfactory performance (Fig. 6). Data series with lower performance have high frequency of low speeds (0-2 m/s), reflecting congested conditions, while data series with higher speeds naturally provided better fits. This could be addressed by using clustered models, in which individual sub-models are estimated on suitable data series with similar characteristics. An approach to do this is presented in the next section.
Figure 5: Histogram of speeds for data series 2, 14 for which satisfactory speed estimation is achieved

Figure 6: Histogram of speeds for data series 10, 16 for which unsatisfactory speed estimation is produced

6 APPLICATION OF CLUSTERED METHOD

The limitation of dealing with heterogeneous data can be addressed by the second methodological approach, presented in Figure 1, which comprises methods such as clustering, loess and classification and allows the adaptation of more flexible and case-specific car following models. In this section, we use data from Naples to verify that this approach could indeed provide better results than the first methodological approach.

First, a model-based clustering is applied to the longest data series (B1695). Traffic states with different characteristics are recognized and data are divided into groups. The factors which are taken into account for the clustering are the speeds of the second and third vehicle \(v_2\) and \(v_3\) and their distance \(D_{23}\), since they are considered as the most relevant for driving behavior according to the preceding analysis. In the clustering algorithm various combinations of models were examined and the optimal number of classes was researched. The BIC index [78] was calculated and the number of classes, which minimizes the index was selected. The classification results for different combinations of models and different number of classes (components) are illustrated in Figure 7. Although the lowest value of BIC index corresponds to 9 classes, the fit on the data is similar for classes between 7 and 9 classes. In addition, even for 4 classes there is not a great loss in relation to the optimal number of classes; therefore the performance of fewer classes could be tested aiming at parsimonious models.
Figure 7: Choice of optimal number of classes

Figure 8: Clustering results for different number of classes
Figure 8 presents the results of clustering for different number of classes. As expected, fewer classes result in simpler clustering, in which case the characteristics of each class are more distinct and easily recognizable. In contrast, the traffic characteristics of each class appear subtler when a greater number of classes (e.g., 6 or 9) is used.

Specific loess models are then calibrated for each traffic state, resulting in a number of models. The other available datasets are then classified into the existing classes created by the B1695 data set. The classification is implemented using the k-nearest neighbor method. Then, the appropriate flexible model is retrieved and applied to the new data for speed estimation.

The expected result would be that the estimation error would be reduced over the previous case, as a higher number of classes would lead to a more precise estimate, because the models are applied to more homogeneous sub-data-sets. On the other hand, in this case less data per group are available and the calibrated models may be too “narrow” to have a good fit to other data, possibly indicating over-fitting. Furthermore, a larger number of classes would lead to difficulties in identifying the distinct underlying behaviors.

The results are summarized in Table 2, indicating that the expected result is not achieved for all datasets. The B1695 data series, which was used for model calibration, provides the best correspondence between the traffic states. For three of the other datasets the best performance was obtained by a single class (and a model with two classes provided very similar results), while for the remaining two the best performance was achieved with four classes. Overall, the clustered approach appears to outperform the simpler loess approach in some cases, and perform similarly in the remaining cases.

<table>
<thead>
<tr>
<th>Data series</th>
<th>Gipps’ model</th>
<th>Loess method</th>
<th>Clustered method (Number of classes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1695</td>
<td>2.70</td>
<td>1.59</td>
<td>1.55 1.53 1.48 1.48 1.46 1.42 1.40 1.37</td>
</tr>
<tr>
<td>C621</td>
<td>6.60</td>
<td>4.34</td>
<td>4.37 4.41 3.99 4.59 4.54 4.10 3.99 5.36</td>
</tr>
<tr>
<td>A358</td>
<td>2.70</td>
<td>2.08</td>
<td>2.10 2.31 3.10 2.34 2.25 2.33 2.33 3.61</td>
</tr>
<tr>
<td>A172</td>
<td>4.60</td>
<td>3.40</td>
<td>3.48 3.06 2.44 3.14 2.85 3.30 3.13 9.39</td>
</tr>
<tr>
<td>C168</td>
<td>2.30</td>
<td>1.78</td>
<td>1.87 2.04 1.95 2.02 2.05 2.09 2.08 2.08</td>
</tr>
<tr>
<td>C171</td>
<td>7.20</td>
<td>6.23</td>
<td>6.31 6.60 7.35 6.50 6.47 8.22 8.19 8.73</td>
</tr>
</tbody>
</table>

Table 2: Results for speed estimation for all Naples data sets using the second methodological approach (t=0.4 s)

7 DISCUSSION AND CONCLUSION

Data driven approaches could be a promising tool for optimization of car-following models, as it may lead to more robust and reliable representation of driving behavior. In this research, an existing methodology for estimation of car following models has been validated to some NGSIM datasets. This simpler approach outperforms the reference (Gipps’) model for all available datasets. The results imply that the proposed methodology outperforms conventional car-following models and could thus provide more accurate representation of car following behavior for simulation purposes. The extended methodology, more elaborate approach, combines clustering, loess and classification, and further improves the performance of the simpler approach in some cases (while providing essentially the same performance as the simpler approach in the remaining cases).
Additional testing on richer data should be performed to determine the factors that determine its performance, as well as develop guidelines for the selection of one or the other approach. Directions of future research should also be outlined. It may be suggested that other microscopic behaviors (lane changing, etc.) to be examined using data-driven methods.

The proposed methodological framework is more flexible, less time-consuming and allows the incorporation of additional parameters that may influence driving behavior (such as drivers’ age, road infrastructure etc.). Resorting cumbersome reformulations of a fixed model form could be impractical. However, conventional models such as Gipps’ model may provide better insight into driving behavior, as they are relied on traffic flow theory. The integration of data-driven methods in traffic micro simulation and the development of Intelligent Transportation Systems could be very helpful. However, additional research and further validation should be conducted.

ACKNOWLEDGEMENT

The authors would like to thank Prof. Vincenzo Punzo from the University of Napoli–Federico II for kindly providing the data collected from Napoli and the FHWA for making the NGSIM data-sets freely available. This research has been supported by the Action: ARISTEIA-II (Action’s Beneficiary: General Secretariat for Research and Technology), co-financed by the European Union (European Social Fund – ESF) and Greek national funds project.

REFERENCES


