Transportation Research Part C 34 (2013) 89–107

Contents lists available at SciVerse ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Dynamic data-driven local traffic state estimation and prediction

Constantinos Antoniou [a],*, Haris N. Koutsopoulos [b], George Yannis [c]

[a] National Technical University of Athens, School of Rural and Surveying Engineering, Zografou 15780, Greece
[b] Division of Transport and Logistics, The Royal Institute of Technology, Stockholm, Sweden
[c] National Technical University of Athens, School of Civil and Environmental Engineering, Zografou 15773, Greece

ARTICLE INFO

Article history:
Received 31 March 2012
Received in revised form 29 May 2013
Accepted 31 May 2013

Keywords:
Traffic state prediction
Local speed prediction
Data-driven approaches
Clustering
Classification
Markov process
Locally weighted regression
Neural network

ABSTRACT

Traffic state prediction is a key problem with considerable implications in modern traffic management. Traffic flow theory has provided significant resources, including models based on traffic flow fundamentals that reflect the underlying phenomena, as well as promote their understanding. They also provide the basis for many traffic simulation models. Speed–density relationships, for example, are routinely used in mesoscopic models. In this paper, an approach for local traffic state estimation and prediction is presented, which exploits available (traffic and other) information and uses data-driven computational approaches. An advantage of the method is its flexibility in incorporating additional explanatory variables. It is also believed that the method is more appropriate for use in the context of mesoscopic traffic simulation models, in place of the traditional speed–density relationships. While these general methods and tools are pre-existing, their application into the specific problem and their integration into the proposed framework for the prediction of traffic state is new. The methodology is illustrated using two freeway data sets from Irvine, CA, and Tel Aviv, Israel. As the proposed models are shown to outperform current state-of-the-art models, they could be valuable when integrated into existing traffic estimation and prediction models.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Traffic state prediction is a key problem with considerable implications in modern traffic management. Several modeling approaches have been used, including Kalman Filter (e.g. Wang et al., 2006a,b; Liu et al., 2006), neural networks (e.g. Vlahogianni et al., 2005; van Lint, 2005, 2008; Vlahogianni et al., 2008; Dunne and Ghosh, 2012) and others (e.g. Stathopoulos and Karlaftis, 2003; El Faouzi et al., 2009). Karlaftis and Vlahogianni (2011) compare statistical methods and neural networks in transportation research, highlighting some of the differences similarities of the two types of data analysis tools. With the emergence of a number of data collection technologies (e.g. c.f. Antoniou et al., 2011, for a review) data-driven approaches offer the potential for the development of approaches that are more appropriate for capturing the dynamic characteristics of traffic. In this paper, an alternative approach is presented, which exploits available (traffic and other) information and data-driven computational approaches to predict local traffic state and speed.

The fundamental traffic flow diagram has been often criticized as restrictive, but has proved useful over the past decades. Hall (1997) and May (1990) provide thorough discussions of traffic stream models, including a number of extensions, such as multi-regime models. Examples include two- and three-regime linear models, and combinations of single regime models (see e.g. Edie, 1961; May and Keller, 1967; Drake et al., 1967).

* Corresponding author. Tel.: +30 210 772 2783; fax: +30 210 772 2629.
E-mail address: antoniou@central.ntua.gr (C. Antoniou).

0968-090X/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trc.2013.05.012

Three somewhat related terms are used in the remainder of this paper; in order to avoid any ambiguity or confusion between them, they are clearly defined here:

- (Traffic) state: the state in which traffic is at any given time can be described by a number of parameters such as flow, density, speed; as these are continuous variables, there can be an infinite number of traffic states.
- Regime: traffic states can be grouped in regimes that reflect a somewhat homogeneous group with similar characteristics; the number of regimes may be determined purely on the basis of mathematical properties of the variables, and the regimes may not have direct interpretations.
- Phase: simple traffic flow theory models assume a small (usually 2 or 3) number of traffic phases, which have direct physical interpretations (e.g. congested or uncongested).

Several papers in the literature have discussed empirical situations in which the fundamental diagram seems to break down. Kerner (2004) has attempted to interpret such empirical observations in terms of a three-phase traffic theory (the three-phase traffic theory includes (i) a free traffic phase, (ii) wide moving jams and (iii) synchronized flow), while new microscopic traffic models that fit the interpretations of three-phase traffic theory (e.g. Kerner and Klenov, 2002) have been developed. It should be noted, however, that there is also vocal criticism to the three-phase traffic theory (e.g. Schoenhof and Helbing, 2009), as it also sometimes fails to fit and explain empirical data.

Various techniques have been used to estimate multi-regime traffic models. For example, Einbeck and Tutz (2004) present an application of multimodal regression to speed-flow data, while Sun and Zhou (2005) use cluster analysis to segment speed–density data and determine the regime boundaries for typical (two-regime and three-regime) speed–density models. Sun et al. (2003, 2004) applied local regression for short term traffic forecasting and report that local regression was superior when compared to nearest neighborhood and kernel smoothing. Toledo et al. (2007) present a local regression approach for processing vehicle position data in order to develop continuous vehicle trajectories and consequently obtain speed and acceleration profiles. The proposed methodology was successfully applied to a set of position data to develop profiles that were subsequently used for the calibration of car-following models. Antoniou and Koutsopoulos (2006b) provide a review of several flexible regression approaches [loess (Cleveland, 1978; Cleveland et al., 1988), support vector regression (SVR) based on support vector machines (Vapnik, 1995, 1998), neural networks (Haykin, 1999)] applied to the task of speed estimation and find that loess behaves better (in terms of accuracy and computational performance) in this context. One particular advantage of the presented approaches (loess, neural networks, and support vector regression) is that, unlike the typical speed–density relationship, they are flexible in incorporating additional explanatory variables, such as time of day, day of week and weather.

Clustering and classification are popular techniques with many applications. El Faouzi (2004) presents a data-driven approach that aggregates multiple estimators, attempting to aggregate all the information which each estimation model embodies (some of which might be lost if only the ‘‘best’’ model was chosen and applied), while El-Faouzi and Lefevre (2006) use two different approaches from evidence theory (classifier fusion and distance-based classification) for clustering and classification for road travel time estimation. Wang et al. (2005) use fuzzy clustering for the classification of car-following behavior into multiple regimes. Azimi and Zhang (2010) apply three different unsupervised learning methods (K-Means, Fuzzy C-Means, and CLARA) to classify freeway traffic flow conditions based on the characteristics of the flow.

The objective of this research is to develop and validate a dynamic data-driven framework that allows for traffic state estimation and prediction. Antoniou and Koutsopoulos (2006a) present a framework for the estimation of speeds using machine-learning approaches. While that work focused on estimation, in the current research the emphasis shifts to traffic state prediction using similar approaches, augmented by additional suitable models, required for the prediction part. Therefore, the main contributions of this paper include the development of a methodology for traffic state prediction. This methodology is based on a set of flexible models, both in terms of functional specification and data to which they can be applied. As the proposed models are shown to outperform current state-of-the-art models, they could be valuable when integrated into existing traffic estimation and prediction models (such as DynaMIT, Ben-Akiva et al., 2002, 2010, DYNASMART, Mahmassani, 2001, or RENAISSANCE, Wang et al., 2006a,b), resulting in more accurate traffic predictions. These predictions could then better support downstream applications, such as traffic guidance generation.

In this paper, the methodology will be presented and results will be provided illustrating how the presented approach performs compared to the existing state-of-the-art. The main components of the methodology and the application setup are presented next, followed by application results and related discussion. Two freeway data sets from Irvine, CA, and Ayalon, Israel, have been used for this research. A discussion section provides further insight and directions for future research.

2. Methodology

2.1. Overall framework

The overall framework is outlined in Fig. 1: the left figure outlines the main methodological components and shows the information flows, while the right figure provides simple examples of the main tasks achieved by each methodological component. In general, each observation may include multiple attributes [e.g. (lagged) speed, density, flow, number of lanes, grade, meteorological information, vehicle mix, driver mix].

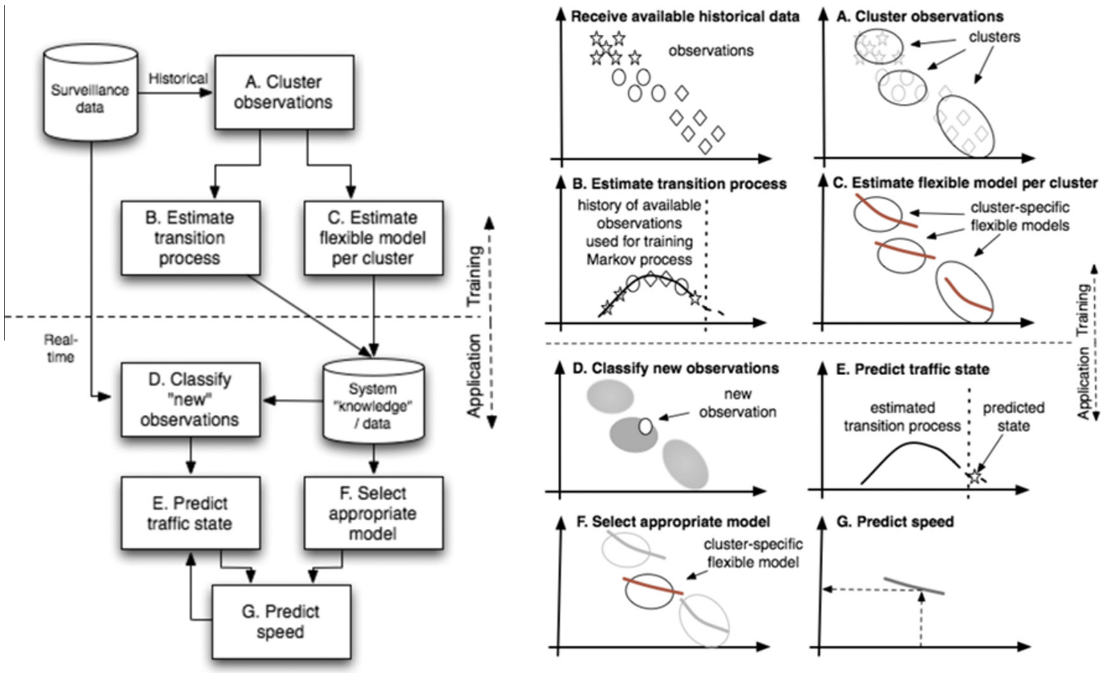

Surveillance data
Historical
A. Cluster observations
B. Estimate transition process
C. Estimate flexible model per cluster
Training
Application
Real-time
D. Classify "new" observations
System "knowledge" / data
E. Predict traffic state
F. Select appropriate model
G. Predict speed
Receive available historical data
observations
A. Cluster observations
clusters
B. Estimate transition process
history of available observations used for training Markov process
C. Estimate flexible model per cluster
cluster-specific flexible models
D. Classify new observations
new observation
E. Predict traffic state
estimated transition process
predicted state
F. Select appropriate model
cluster-specific flexible model
G. Predict speed

Fig. 1. Overall local traffic state prediction framework.

The methodology comprises training and application steps. During the training step archived surveillance data are used to (A) identify the various traffic states through clustering the available observations; (B) estimate the transition processes between these regimes; and (C) estimate cluster-specific traffic models. This information is stored into a knowledge base and supports the application of the framework.

As new measurements become available, they are (D) classified into the appropriate regimes and – based on the transition processes and the short-term evolution of the traffic state – (E) short-term predictions of the traffic state are performed using the applicable estimated transition processes. Furthermore, (F) the appropriate flexible traffic model is retrieved and applied to the new observations to (G) perform speed predictions.

The framework presented in Fig. 1 comprises a number of data driven models, outlined next and presented in more detail in the following subsections. While these general methods and tools are pre-existing, their application into the specific problem and their integration into the proposed framework for the prediction of traffic state is new.

2.1.1. Clustering and classification

The available observations have different characteristics that can be used to cluster them (A) into groups with similar characteristics. Clustering is a well-researched area with a large number of available approaches and algorithms, often based on heuristics. Clustering involves several decisions, such as the optimal number of clusters that effectively clusters the observations to meaningful clusters. Conflicting objectives characterize this task, as on the one hand a larger number of clusters may provide a more precise clustering, while a smaller number of clusters provide a more manageable (and possibly easier to interpret) clustering. When new observations become available, they can be classified (D) into one of the available clusters based on their specific attributes.

2.1.2. Modeling the evolution of traffic states

The classified observations result in a time series of clusters. Studying the evolution of this time-series provides the ability to predict the future state, based on the last few states, through the estimation of an appropriate state-predictive process (B). For example, assuming that the states identified under clustering are A, B, C, the model should be trained to predict that, given the sequence of the last few states is A–A–B the next state is C. In general, simpler processes, with fewer states are expected to have a lower rate of misclassification.

2.1.3. State-specific speed prediction

This step (C) employs appropriate flexible regression models based on the observations belonging to the corresponding cluster. These cluster-specific models (C) relate speed to all relevant data (e.g. density, flow, number of lanes, grade, meteorological information, vehicle mix, driver mix). When the cluster of a future observation has been predicted (E), the appropriate function for that cluster can be selected (F) and used to make a speed prediction (G). Such richer, data-driven models, are statistically driven (always motivated by traffic flow theory principles), and they have the potential to provide accurate speed predictions, as they incorporate more diverse data as explanatory variables.

2.2. Clustering and classification

2.2.1. Model-based clustering

Clustering and classification are tasks that are rather well researched as they have extensive applications in many practical and research fields. As a result, a range of approaches and algorithms are available, often based on heuristics. One rigorous approach to cluster analysis is based on probability modes (see Bock, 1998a,b, for a survey). Some of the most popular heuristics used for clustering are approximate estimation methods for appropriately defined probability models (Fraley and Raftery, 2002). For example, standard k-means clustering (Mitchell, 1997) is equivalent to known procedures for approximately maximizing the multivariate normal classification likelihood when the covariance matrix is the same for each component and proportional to the identity matrix (Fraley and Raftery, 2002).

Finite mixture models have been proposed and studied in the context of clustering (Wolfe, 1970; Edwards and Cavalli-Sforza, 1965; Day, 1969; Scott and Symons, 1971; Duda and Hart, 1973; Binder, 1978), often as a statistical approach to shed some light into practical questions that arise from the application of clustering methods (McLachlan and Basford, 1988; Banfield and Raftery, 1993; Cheeseman and Stutz, 1995; Fraley and Raftery, 1998). Each component probability distribution in finite mixture models corresponds to a cluster. The problems of determining the number of clusters and of choosing an appropriate clustering method can be recast as statistical model choice problems, and models that differ in numbers of components and/or in component distributions can be compared. Outliers are handled by adding one or more components representing a different distribution for outlying data.

Given data $\mathbf{y}$ with independent multivariate observations $y_1,\ldots,y_n$ the likelihood for a mixture model with G components is:

$$L_{MIX}(\theta_1,\ldots,\theta_G;\tau_1,\ldots,\tau_G|\mathbf{y}) = \prod_{i=1}^{n}\sum_{k=1}^{G}\tau_k f_k(\mathbf{y}_i|\theta_k) \tag{1}$$

where f_k and θ_k are the density and parameters of the kth component in the mixture and τ_k is the probability that an observation belongs to the kth component.

Data generated by mixtures of multivariate normal densities are characterized by groups or clusters centered at the means, with ellipsoidal surfaces of constant density. The geometric features (shape, volume, orientation) of the clusters are determined by the covariances Σ_k, which may be further parameterized to impose cross-cluster constraints. In the simplest case of spherical clusters of the same size $\Sigma_k = \lambda I$, while in the case of clusters with the same geometry (but not necessarily spherical) $\Sigma_k = \Sigma$ (Friedman and Rubin, 1967). Only one parameter is needed to capture the covariance structure of the mixture when $\Sigma_k = \lambda I$, while for d-dimensional data $d(d+1)/2$ and $G(d(d+1)/2)$ parameters are needed for constant Σ_k and unrestricted Σ_k (Scott and Symons, 1971). Banfield and Raftery (1993) and Murtagh and Raftery (1984) proposed more flexible and general frameworks for geometric cross-cluster constraints in multivariate normal mixtures by parameterizing covariance matrices through eigenvalue decomposition.

The purpose of cluster analysis is to classify data of previously unknown structure into meaningful groupings. A strategy for cluster analysis based on mixture models is outlined next (Fraley and Raftery, 2002). The strategy comprises three core elements: (i) initialization via model-based hierarchical agglomerative clustering, (ii) maximum likelihood estimation via the expectation–maximization (EM) algorithm, and (iii) selection of the model and the number of clusters using approximate Bayes factors with the BIC (Bayesian Information Criterion) (Schwarz, 1978) approximation.

Model-based hierarchical agglomerative clustering is an approach to computing an approximate maximum for the classification likelihood:

$$L_{CL}(\theta_1,\ldots,\theta_G;\ell_1,\ldots,\ell_n|\mathbf{y}) = \prod_{i=1}^{n} f_{\ell_i}(\mathbf{y}_i|\theta_{\ell_i}) \tag{2}$$

where ℓ_i are labels indicating a unique classification of each observation, taking the value k if $\mathbf{y}_i$ belongs to the kth component. In the mixture likelihood (Eq. (1)), each component is weighted by the probability that an observation belongs to that component. The presence of the class labels in the classification likelihood (Eq. (2)) introduces a combinatorial aspect that makes exact maximization impractical (Fraley and Raftery, 2002).

Murtagh and Raftery (1984) successfully applied model-based agglomerative hierarchical clustering to problems in character recognition using a multivariate normal model, with volume and shape held constant across clusters. This approach was generalized by Banfield and Raftery (1993) to other models and applications.

In hierarchical agglomeration, each stage of merging corresponds to a unique number of clusters and a unique partition of the data. A given partition can be transformed into indicator variables, which can then be used as conditional probabilities in an M step of EM for parameter estimation, initializing an EM algorithm. This, combined with Bayes factors as approximated by BIC for model selection, yields a comprehensive clustering strategy:

- Determine a maximum number of clusters, M, and a set of mixture models to consider.
- Perform hierarchical agglomeration to approximately maximize the classification likelihood for each model and obtain the corresponding classifications for up to M groups.
- Apply the EM algorithm for each model and each number of clusters $2,\ldots,M$, starting with the classification from hierarchical agglomeration.
- Compute BIC for the one-cluster case for each model and for the mixture model with the optimal parameters from EM for $2,\ldots,M$ clusters.

Once the optimal number of clusters has been determined based on the outlined approach, it is possible to consider smaller numbers of clusters, leading to simpler models.

2.2.2. Nearest neighbors classification

Clustering methods are usually accompanied by a classification algorithm so that they can be applied. Nearest neighbor classification is one of the standard classification methods. During the application phase, standard methods, such as the k-nearest neighbor approach, can be used to classify new traffic measurements (characterized by e.g. flow and density) to the most appropriate cluster. k-nearest neighborhood learning is the most basic instance-based method, and assumes that all instances (or observations) correspond to points in the n-dimensional space (Mitchell, 1997). The nearest neighbors of an instance are defined in terms of the standard Euclidean distance.

Let an observation x be described by the feature tuple $\langle a_1(x), a_2(x), \ldots, a_n(x)\rangle$ where $a_r(x)$ denotes the values of the rth attribute of x. In the context of traffic dynamics, the attributes of x could be density and flow, but also other parameters, such as time of day, prevailing weather conditions, and traffic mix. The distance between two instances x_i and x_j is then defined to be:

$$d(x_i, x_j) \equiv \sqrt{\sum\nolimits_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{3}$$

In nearest-neighbor learning the target function may be either discrete-valued or real-valued. In the discrete case, such as this one, where the goal is to assign each new instance x_q to a cluster, the algorithm selected the k instances from the training set that are nearest to x_q (as defined by the distance above), and returns

$$\hat{f}(x_q) \leftarrow \arg\max_{u\in V} \sum_{i=1}^{k} \delta(u, f(x_i)) \tag{4}$$

where δ is the Kronecker operator

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The special case in which a single neighbor is considered (i.e. $k = 1$) means that the class of each new observation is predicted to be the class of the closest training sample (nearest-neighbor).

2.2.3. Classification using neural networks

Classification was also performed using neural networks, in order to provide a reference case using a well-established technique. Neural networks (c.f. e.g. Haykin, 1999; Ripley, 1996) have been presented in many traffic related applications (e.g. Vlahogianni et al., 2005). Neural networks are well-described in the literature and therefore not presented here in the interest of economy of space. It is indicated, however, that single-hidden-layer neural networks are considered in this context.

2.3. Modeling the evolution of traffic states

One of the most general models for a stationary categorical process taking values in a finite categorical space X, is a full Markov chain (of possibly high, but finite order – for a Markov chain the order p is the number of past states on which a future state depends) (Markov, 1971). For example, in a traffic flow theory context, traffic flow might be categorized as one of four states (say A, B, C and D), defining the categorical space X. A stationary full Markov chain of order p exists whenever the transition mechanism has no specific structure; that is the state space is the entire X_p. While such general models may be theoretically attractive, they also have practical limitations. For example, a full Markov chain is rather inflexible in terms of the number of parameters (i.e. nodes) that it can represent. For a model with 4 states (as in the simple example of A through D above), chains with 0–5 parameters have dimensions of 3, 12, 48, 192 and 768, respectively. Markov chains can only be fitted in these "intervals", thus reducing the model flexibility (e.g. if 48 parameters are not enough, one needs to estimate 192 parameters; intermediate values are not possible). This issue introduces another problem with the full Markov chain model, the "dimensionality curse", as the dimension of the model increases exponentially with the order p.

Markov processes have found many applications in a diverse number of fields. For example, Geroliminis and Skabardonis (2005) propose an analytical methodology for prediction of the platoon arrival profiles and queue length along signalized arterials using Markov decision processes, while Yeon et al. (2008) develop a model that can estimate travel time on a freeway using Discrete Time Markov Chains (DTMC) where the states correspond to whether or not the link is congested). Other applications of Markov processes in transport-related literature include indicatively pavement management (Abaza et al., 2004) and bridge maintenance management (Scherer and Glagola, 1994; Ortiz-Garcia et al., 2006). Stamoulakatos and Sykas (2007) model mobile terminals communication with their base station using hidden Markov models in combination with clustering algorithms.

Variable length Markov chains (vlmc) address both issues introduced above (inflexibility in terms of dimension and lack of scalability) and provide a natural and elegant way to avoid (some of) the difficulties mentioned. The idea is to allow the memory of the Markov chain to have a variable length, depending on the observed past values (Maechler and Buehlmann, 2004). For example, while a past history of states A–A–B–C–C may be a good indication of a next state being D, for other cases it might be sufficient to "store" a sequence of A–C as a precursor to a state C. The first example would indicate that the order p of the Markov chain would be five, resulting in a dimension of 768. However, a history of A–C is sufficient to make another state prediction and therefore, the "tree" of history following that tree is not required and can be deleted (or pruned, as is said in this context). A memory of variable length (five in the first case, but only two in the latter) is thus sufficient.

Using this idea, fitting a vlmc from data involves estimation of the structure of the variable length memory, which can be reformulated as a problem of estimating a tree, using the so-called context algorithm (Rissanen, 1983), which can be implemented very efficiently. In the fitted tree-structured models, every terminal node (as well as some internal nodes) represents a state in the Markov chain and is equipped with corresponding transition probabilities. The context algorithm grows a large tree and prunes it back subsequently. The pruning part requires specification of a tuning parameter, the so-called cutoff. The cutoff K is a threshold value when comparing a tree with its subtree by pruning away one terminal node; the comparison is made with respect to the difference of deviance from the two trees. A large cutoff has a stronger tendency for pruning and yields smaller estimated context trees, i.e. a smaller dimension of the model.

2.4. State-based speed prediction

Antoniou and Koutsopoulos (2006b) review several flexible regression approaches, such as locally weighted regression, loess (Cleveland, 1978; Cleveland et al., 1988), support vector regression (SVR) based on support vector machines (Vapnik, 1995, 1998), and neural networks (Haykin, 1999) applied to the task of speed estimation. They report that loess behaves better in this context. Based on this analysis, loess is considered in this methodology as an example of flexible regression approaches.

Locally weighted regression (loess) was first proposed by Cleveland (1978), and Cleveland et al. (1988). Cleveland and Devlin (1988) report various application areas of the method, such as support for exploratory graphical data analysis, provision of additional regression diagnostics for testing parametric models fitted to the data, and direct use of the local regression functions in place of parametric functions. The method has also found applications in machine learning. It is used as a form of memory (or instance)-based learning, to learn continuous nonlinear mappings in applications such as learning robot dynamics, and process models (Atkenson et al., 1997).

Locally weighted regression can also be viewed as a generalization of the k-nearest neighbor approaches (Mitchell, 1997). Unlike the k-nearest neighbor approach, however, which can be thought of as approximating a target function $g(x)$ at a single point, locally weighted regression constructs an explicit approximation of $g(x)$ over a local region surrounding this point. Several functional forms can be used for this approximation, including e.g. a linear or a quadratic function, or a multilayer neural network.

Following Cleveland and Devlin (1988), locally weighted regression $y_i = \hat{g}(x_i) + \varepsilon_i$, where ε_i are residual errors, provides an estimate $\hat{g}(x)$ of the regression surface at any value x in the p-dimensional space of the independent variables. In this case, y_i is local speed and independent, explanatory variables might include (possibly lagged) flow and density, geometric characteristics, vehicle mix or prevailing weather conditions. Let q be an integer, $1 \leqslant q \leqslant n$. The estimate of $g(\,)$ at x uses the q observations whose x_i values are closest to x. Each of these points is weighted according to its distance from x, with points close to x having large weights and points farther from x having small weights. A function of the independent variables is fitted to the dependent variable using least squares with these weights. $y = \hat{g}(x)$ is then the value of this fitted function at x. The objective function to be minimized is the weighted sum of square residuals $\sum_{i=1}^{q} w_i \varepsilon_i^2$.

A commonly used weight function is the tricube weight function (Cleveland and Devlin, 1988):

$$W(u) \equiv \begin{cases} (1-u^3)^3 & \text{if } 0 \leqslant u < 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Let $d(x)$ be the distance of the qth-nearest x_i to x. Then, the weight for observation (y_i,x_i) is (Cleveland and Devlin, 1988):

$$w_i(x) = W(p(x,x_i)/d(x)) \tag{7}$$

where $p(x,x_i)$ is a distance function in the space of the independent variables. In applications with a single independent variable p is the Euclidean distance, while in the multivariate case variables may have to be scaled first (e.g. by their corresponding standard deviation).

An efficient computational approach for estimating the parameters of the local regressions is presented in Cleveland et al. (1988).

3. Case study

3.1. Data and experimental design

Two freeway data-sets from Irvine, CA, and Tel Aviv in Israel have been used for this research. In both cases, weekday data were used. The Irvine data set includes five days of sensor data from freeway I-405. The application involved training/calibration with four days of data and subsequent testing/validation of the model framework for the fifth day (not used in the calibration). Data from 10 am to 12 midnight have been used, since this period includes the (pm) peak flow for this direction. Speed, occupancy and flow data over 2-min intervals were available for calibration and validation. Occupancy data have been converted to density using a relationship from May (1990, Eq. (7.2) in p. 193).

The second data set was collected in Highway 20 (Ayalon Highway), a major intra-city freeway running through the center of Tel Aviv in Israel. Four days of data were used for the training of the models and a different fifth day was used for validation. Speed, occupancy and flow data were available and were aggregated over 5-min intervals. Occupancy data have been converted to density using the same relationship as above. Furthermore, it is noted that the occupancy data for Ayalon were rounded to percentage points, which – while common practice – could lead to rounding errors when the occupancy values are very small.

The following different cases are developed, based on the type of approach that is used for state (where applicable) and speed prediction:

I. Typical speed–density relationship: A commonly used speed–density relationship is fit to the speed and density data of the training data set. The estimated relationship is then used to calculate speed values based on the densities in the test data set. The true densities (instead of predicted) are used in this process, thus eliminating any prediction error and providing an even better than expected prediction of speeds for this baseline model.
II. Locally weighted regression: The locally weighted regression (loess) algorithm is used to fit a flexible curve predicting speed based on density and flow measurements, using the training data-set. The test dataset is then used to predict the speed using density and flow measurements. The difference from the previous case, besides the more flexible functional form, is that this approach can easily be extended with additional explanatory variables, something that is not easy (if at all possible) using the typical speed–density relationship.
III. Proposed framework: The complete state and speed prediction framework presented (Fig. 1) has been applied using the available data. In this scenario, a sensitivity analysis was also conducted to assess the impact of the number of clusters:
    a. The optimal number of states, i.e. the number of states that minimizes the BIC, based on the results from the model-based clustering algorithm.
    b. Smaller number of clusters: The main difference from the previous case is that a smaller number of clusters has been used in an attempt to assess what the impact of a more parsimonious model structure would be on the model ability to predict traffic conditions. Using a smaller number of clusters one can argue that it might be now easier to compare this approach with the two or three-phase traffic models.
IV. Simplified proposed framework: The complete state and speed prediction framework is used, but neural networks are used for the clustering and classification steps. This is a simpler approach that is implemented in order to assess the incremental benefits of the proposed framework components.

The following speed–density relationship model was used as the reference model (Ben-Akiva et al., 2010):

$$u = \begin{cases} u_f & \text{if} \quad k < k_{min} \\ u_f[1 - ((k - k_{min})/k_{jam})^{\beta}]^{\alpha} & \text{otherwise} \end{cases} \tag{8}$$

where u denotes the space mean speed, u_f the free flow speed, k the density, k_{min} the minimum density, k_{jam} the jam density, and α and β model parameters. This is a variant of the speed–density traffic flow theory relationship that is commonly used in mesoscopic traffic simulation models. For example, this is the relationship used in the DynaMIT model (Ben-Akiva et al., 2010) and very similar to the relationship used in the DynaSMART (Mahmassani, 2001) and mezzo (Burghout et al., 2005) models.

The performance of the presented approaches is assessed using a number of appropriate goodness-of-fit statistics (e.g. Toledo and Koutsopoulos, 2004; Hollander and Liu, 2008). The purpose of using multiple statistics is that they can capture different aspects of the obtained results. The normalized root mean square error, RMSN, and root mean square percent error RMSPE (Pindyck and Rubinfeld, 1997) quantify the overall error of the method, while the mean percent error, MPE (Pindyck and Rubinfeld, 1997) indicates the existence of systematic under- or over-prediction. Another measure that provides information on the relative error is Theil's inequality coefficient (Theil, 1961). U is bounded and takes values between zero and one ($0 \leqslant U \leqslant 1$, where $U = 0$ implies perfect fit between observed and modeled measurements). Theil's inequality coefficient may be decomposed into three proportions of inequality: the bias (U^M), the variance (U^S) and the covariance (U^C) proportions. By definition, the three proportions sum to 1 ($U^M + U^S + U^C = 1$). The bias proportion reflects the systematic error. The variance proportion indicates how well the simulation model is able to replicate the variability in the observed data. These two proportions should be as close to zero as possible. The covariance proportion measures the remaining error and therefore should be close to 1.

3.2. Model training

The fitting of the locally weighted regression is performed – as the name suggests – locally. That is, the fit at point x, is made using points in a neighborhood of x, weighted by their distance from x. The number of points that are considered has a direct impact on the smoothness of the regression curve (with more points resulting in a smoother line). The parameter which controls the degree of smoothing of the regression line (parameter α, also referred to as span) in the locally weighted regression (loess) was determined using a line search for the case where a single curve is estimated for the entire data set, or a grid-search (along with the number of k neighbors to be considered for the k-nearest neighborhood classification algorithm) in the case where cluster-specific curves are fit. The range of estimated values for the smoothing parameter α are about 0.5–0.6 for the single curve and between 0.9 and 1.9 for the cases with multiple clusters (with a higher number of clusters usually resulting in a larger value of the smoothing parameter). Larger values of α result in smoother functions that are less affected by fluctuations in the data (e.g. an outliner due to a measurement error in a detector). Smaller values of α result in curves that follow the individual data points more closely.

Using the grid-search approach mentioned above, the number of k-nearest neighbors is found to be between 2 and 8, with higher values being obtained when fewer clusters are used.

It is worth reiterating that these values have been obtained using line- or grid-search techniques and selecting the values that minimized the objective functions (as indicated by the measures of effectiveness presented in the previous section; it is possible that different measures of effectiveness would result in somewhat different values and this would be an interesting point to consider in future research). It would be interesting to explore the impact of choosing different parameters that might have resulted in a marginally lower value of the objective function, but might provide superior results.

The implementation of this research was performed within the R Software for Statistical Computing v.2.15.1 (R Development Core Team, 2012; Venables and Ripley, 2002) using the Mclust package (Fraley and Raftery, 2002) for model-based clustering, the vlmc package (Maechler and Buehlmann, 2004) for estimation of variable-length Markov chains and the nnet package (Ripley, 1996) for classification using neural networks.

3.2.1. Clustering

The best functional form of the mixtures to be considered for clustering, and the optimal number of clusters were sought using the model-based clustering algorithm (Fraley and Raftery, 2002). A large number of different mixture models were considered, varying in terms of shape and volume. For example, two clusters may have the same volume, shape and orientation; each of these restrictions (volume, shape and orientation) could also be relaxed to lead to a more flexible model. The less restrictive model, having an ellipsoidal structure with no restrictions on the volume, shape and orientation among clusters, is expected to match the data at hand better.

Fig. 2 shows – as an example – the clustering results (using the procedure outlined in Section 2.2.1) for the Irvine network for some cases:

- Four clusters using the most restrictive model (same volume, shape and orientation).
- Four clusters using the most flexible model.
- Three clusters using the most flexible model.

Several observations can be based on this figure. For example, the restrictions on the volume, shape and orientation of the clusters in the top subfigure limit the ability of the clusters to adequately reflect the shape of the data. As a result of these restrictions, the clusters cannot be formed in a way that is consistent with traffic flow theory. For example, the higher-density region of the data is not captured at all; furthermore, the clusters that have been created cannot be behaviorally explained. Relaxing these restrictions, however, allows the clusters to be formed in ways that are more meaningful and consistent with traffic flow theory. The middle subfigure of Fig. 2 shows how the same number of clusters (four) can be used – with the relaxation of the shape, volume and orientation restrictions – to better capture the data points. The bottom subfigure demonstrates how a smaller number of clusters (in this case three, instead of four) can still adequately capture the data structure, in this case by essentially re-arranging the data in the higher-density clusters.

Fig. 3 shows the obtained curves of fit for each number of clusters. “Final” refers to the fit obtained by the selected, best model (in terms of volume, shape and orientation of the clusters), while “Reference” refers to the values obtained using the most restrictive model (equal volume, shape and orientation for all clusters). The “optimal” model has 4 clusters for Irvine and 8 clusters for Ayalon. However, it is recognized that determining the number of clusters solely on the basis of a goodness-of-fit measure such as BIC is likely to favor larger numbers of clusters, which might also be more difficult to interpret from a traffic flow theory point of view. Furthermore, considering the incremental benefits of additional clusters (resulting in more complicated traffic descriptions), one notices that a smaller number of clusters appears to give a fit close to the optimal.

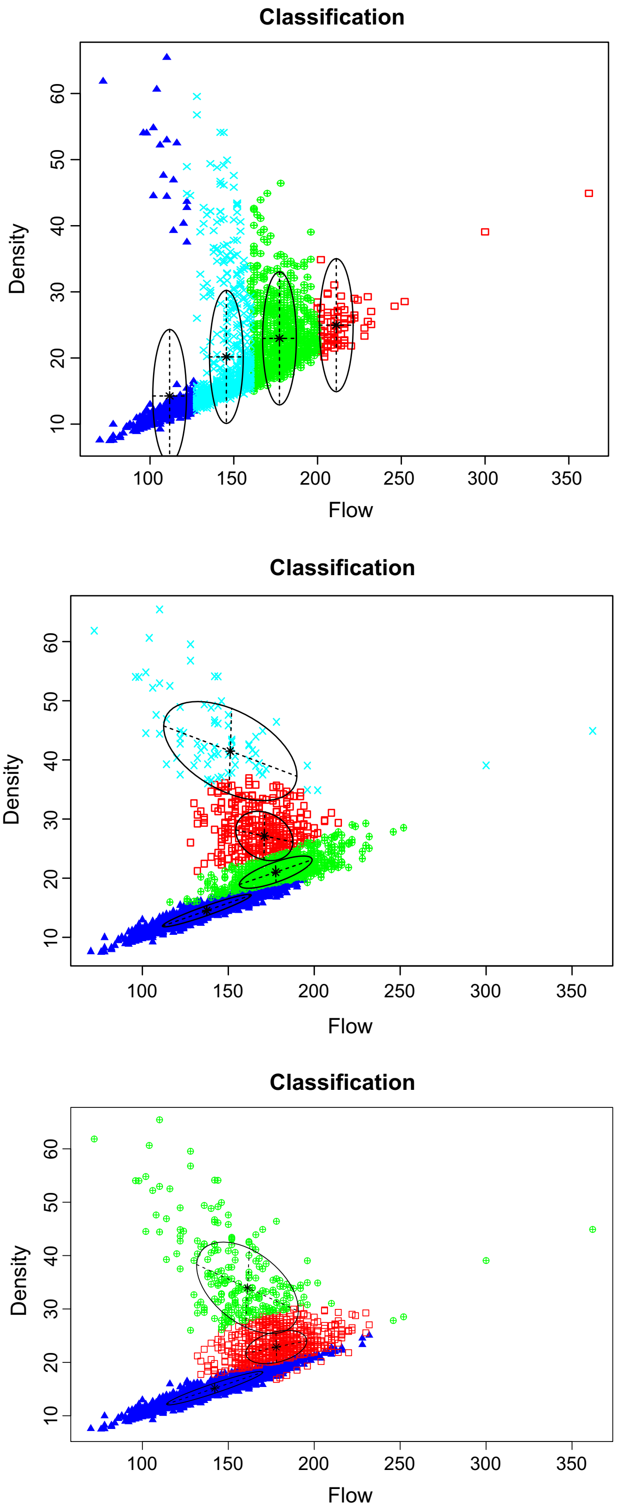


Fig. 2. Different clustering scenarios applied on the Irvine dataset.

For example, 5 clusters provide a similar fit to the optimal number of 8 clusters (in the Ayalon dataset). Furthermore, when the number of clusters is further reduced, e.g. down to 3 clusters, then the loss may not be large. In the remainder of this paper up to three cases per application will be investigated in parallel, i.e. the optimal number of clusters (based on the

BIC criterion, as shown in Fig. 3, i.e. 4 for Irvine and 8 for Ayalon) as well as numbers of clusters, resulting in more parsimonious models (in particular 5 and 3 clusters). In the remainder of this paper, "optimal" number of clusters refers to the number of clusters that minimizes the value of BIC.

Fig. 4 provides a visual representation of the clustering results for different numbers of clusters for the two networks. The results for the optimal clustering are presented in the top, followed by the results for the more parsimonious models. As expected, a smaller number of clusters results in a simpler clustering that could be more easily interpreted into recognizable and distinct traffic states. It is interesting to note that the resulting sets of clusters in the two networks have similar geometries. This becomes particularly evident as the number of clusters decreases, and especially in the 3-cluster cases.

3.2.2. Markov chain training

As mentioned in Section 2.3, the difference between a variable length Markov chain (vlmc) and a complete Markov chain model is that in the former, the "branches" that are not needed are deleted or pruned. In practice, the way the vlmc works is that it computes a huge tree and then prunes it, based on some appropriate parameter value. The cutoff for this pruning is an input parameter, which – in the case of this research – was obtained using a line-search for the value that provided the lowest value for the Akaike Information Criterion (AIC, Akaike, 1974). The "objective function" in this case was the degree to which the resulting vlmc was capable to reproduce the history. The results of this sensitivity analysis are presented in Fig. 5. For each data set, the sensitivity analysis with respect to the cutoff K value is presented for different numbers of clusters. For the Irvine dataset using four clusters, the optimal value for the cutoff parameter is 4.1. The minimum AIC obtained with this cutoff value is equal to 1984. Similarly, for the Ayalon freeway and eight clusters the optimal cutoff value is equal to 7.6 (minimum AIC equal to 1299.8). When the number of clusters is restricted to 5 for the Irvine data, then the optimal cutoff value is 4.6 (minimum AIC equal to 1985) and when only three clusters are considered the cutoff value drops to 3.7 (minimum AIC equal to 1389). Similarly, for the Ayalon data the optimal cutoff value for 5 clusters is 6.2 (minimum AIC equal to 859.9), while for the case of 3 clusters the optimal cutoff is equal to 2.8 (minimum AIC value equal to 479.1). A smaller number of clusters leads to a smaller model and therefore a lower value of the cutoff.

Fig. 6 illustrates the residuals of the fitted variable-length Markov Chain models against the contexts, i.e., produces a boxplot of residuals for all contexts used in the model fit. Intuitively, a context is a "case" that is not pruned from the context tree. For example, if 331 is retained as a context (and shown as a tick in the x-axis), it means that a sequence of states 3, 3 and then 1 can be used to predict the next state with reasonable accuracy. On the other hand, sequences of states that are not systematically followed by a specific state (and therefore cannot be used effectively for prediction) are not retained. The number of observations per context state is also illustrated, above the x-axis of the figure. Furthermore, the width of each boxplot is proportional to the square root of the number of observations that it represents. A small number of sequences implies that the evolution of the traffic state is usually explained by the preceding state. Examining the retained contexts, one observes that when a smaller number of states is used (4 for the Irvine data set), then longer sequences of states are retained as contexts. This is reasonable and expected, since when a smaller number of states is available, the same sequence of states is more likely to be observed (and similarly, the chance that the next state will be correctly predicted is increased, as there are fewer states). This is observed also when comparing the same plots for the Ayalon data set with 5 and 3 clusters; these plots however are not shown here in the interest of space economy.

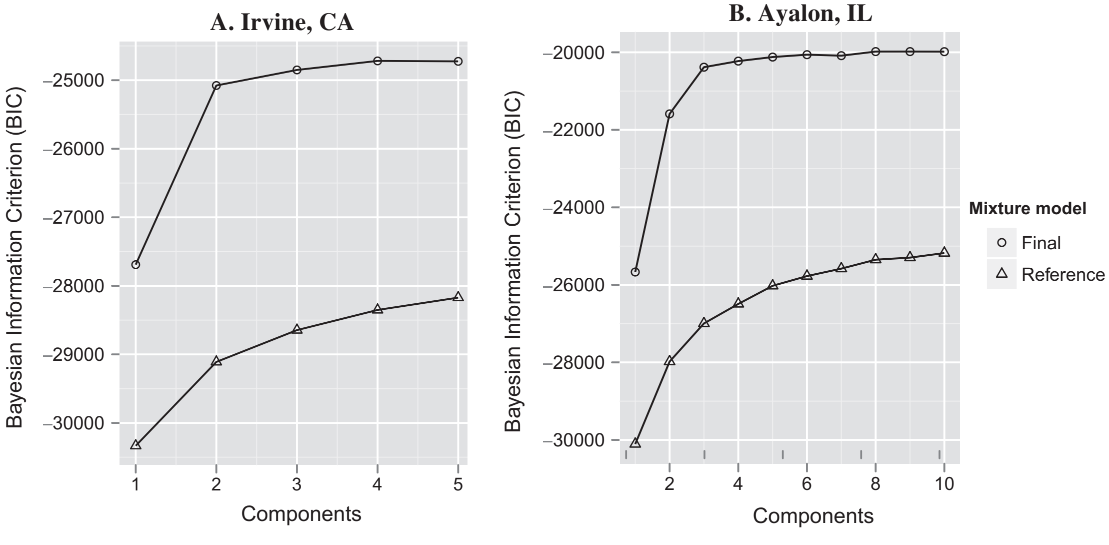
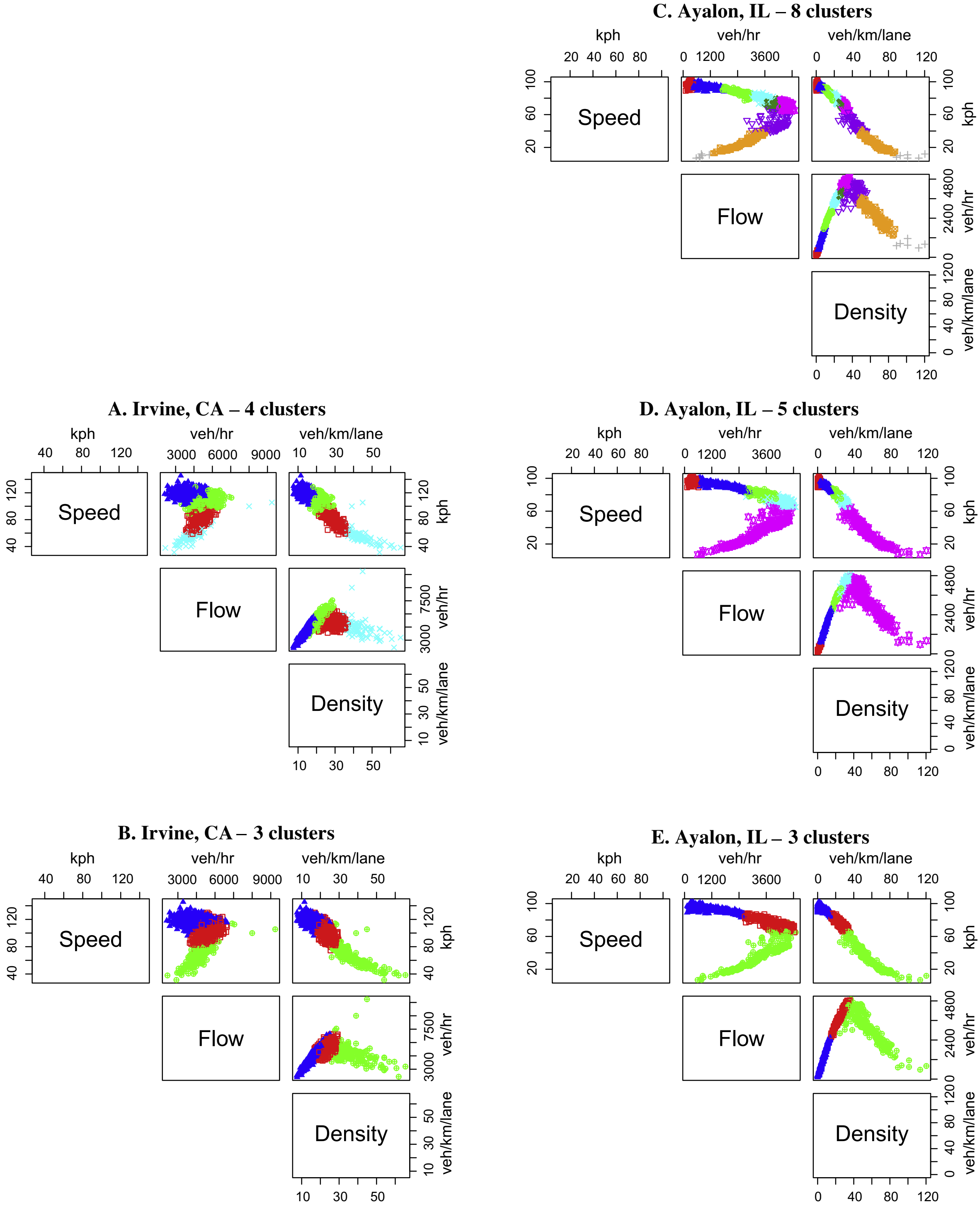


Fig. 3. Optimal number of clusters (left: Irvine, right: Ayalon).



C. Ayalon, IL – 8 clusters

A. Irvine, CA – 4 clusters

D. Ayalon, IL – 5 clusters

B. Irvine, CA – 3 clusters

E. Ayalon, IL – 3 clusters

Fig. 4. Visualization of clustering results for different number of clusters (left: Irvine, right: Ayalon).

Fig. 7 presents the context tree for the estimated models, which provides an alternative way to interpret the transition probabilities. Consider the tree for Ayalon and eight clusters, which – due to the larger number of clusters – has a simpler form. If the previous cluster is 2 through 8, then there is a fairly good idea of what the next cluster will be. Let's consider now the case where the previous observation falls in cluster 1. Then, if the observation before that is 3, then one set of transition probabilities is obtained, while if, on the other hand, the previous observation (before 1) was anything except for 3, then a different set of transition probabilities is used. When the number of clusters is reduced (as shown in the bottom right

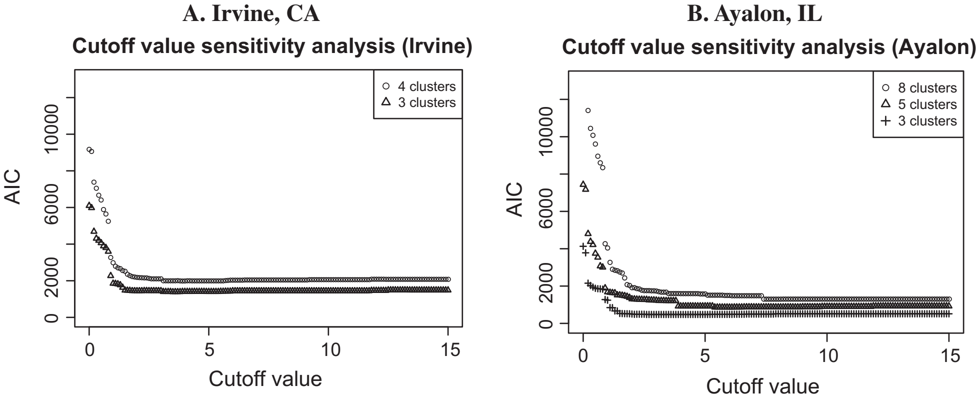


Fig. 5. Sensitivity analysis for the determination of the optimal variable length Markov chain process (left: Irvine, right: Ayalon).

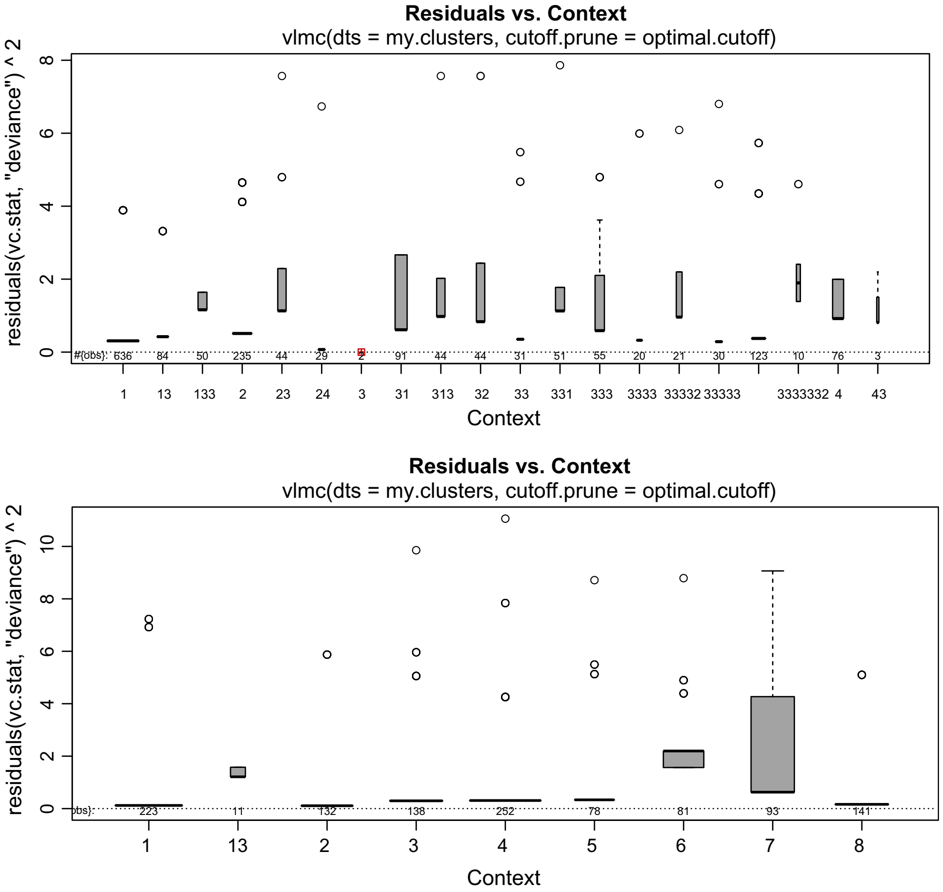


Fig. 6. Plots of residuals vs. context (case retained in the variable length Markov chain) (top: Irvine, bottom: Ayalon).

subfigure, where the number of clusters for the same Ayalon data set has been reduced to 5), the structure of the tree becomes richer, as the smaller number of clusters forces (or allows) more elaborate combinations of cluster histories. The more complicated context tree for the Irvine network and four clusters only can be interpreted in a similar way.

3.3. Application for speed prediction

Speed prediction for one interval ahead is considered. The duration of the interval is consistent with the time over which the surveillance data were aggregated, i.e. 2 min for Irvine and 5 min for Ayalon.

3.3.1. Traffic state prediction

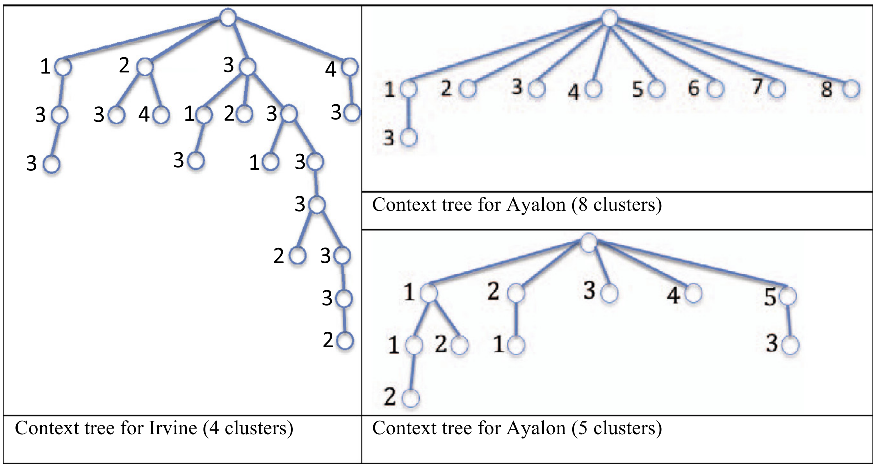
Traffic state prediction involves several steps utilizing different data. Clustering and classification use traffic data (i.e. density and flow), while state prediction relies on the traffic state during the previous intervals. In the presented framework in this research, clustering is performed using model based clustering. Furthermore, two different options for implementing the subsequent steps of the process outlined in Fig. 1 are considered. In the first process, classification and one-step prediction are performed using single-hidden layer neural networks (with two units in the hidden layer), while in the second, classification is performed using *k*-nearest neighbors and one-step prediction using variable length Markov chains. In each case, the one-step predicted state with the highest predicted probability was selected as the most likely future state. The same explanatory variables (flow and density) were used for the prediction in both cases (vlmc and neural net).

Using the calibrated Markov chain or neural networks to predict the future traffic state involves the inherent danger of misclassification, i.e. predicting a state that is different than the state that would be determined if the actual data from the future observation were classified using the *k*-nearest neighbor algorithm or the neural network. The extent of misclassification can affect the accuracy of the overall methodology. In this section (Table 1), the one-step predicted cluster is compared with the "true" cluster, i.e. the cluster in which that observation would be classified when the data (flow, density) becomes available. It is noted that the "true" cluster is the cluster in which the observation is assigned when – in the future – the true measurements become available. Therefore, this depends on the clustering and classification methodology. For example, in the Ayalon data-set with 5 clusters, the *k*-nearest-neighbors method has assigned 83 observations in the "true" cluster 3, while the neural networks method has assigned 54. This clarification is useful in understanding that the distinction between the "true" and predicted classification is essentially the type of information that is available (measured vs. predicted).

For the Irvine dataset, with the optimal number of 4 clusters, the Markov chain prediction correctly classified the traffic states in 79% of the cases, while the neural network predicts correctly 82% of the cases. The following table provides some more detail on this classification, with correct classifications shown in the diagonal and in bold. Cells with zero observations have been left blank for clarity. When the number of clusters is reduced to 3, then the number of correctly classified states reaches 84% of considered states for the Markov chain and 86% for the neural network, which is intuitive, considering that with a smaller number of states, there is a lower chance of misclassification.

For the Ayalon dataset, when the optimal number of (8) clusters is considered, 85% of the states are correctly classified with the Markov chain prediction and 78% are correctly classified when the neural network approach is used. When the number of clusters is limited to 5, the percentage of correctly classified states increases to 90% for the Markov chain prediction approaches and 84% for the neural network approach, while when only three clusters are considered, 92% of the observations are classified in the correct state with the *k*-nearest neighbors (knn) and Markov chain approach, and 88% when the neural networks approach is used. Again, a better predictive performance is expected when a smaller number of candidate clusters is considered.

One question that arises from the difference in the classification performance between the two data sets is why is performance better with the Ayalon data set. Looking at Fig. 4, the range of observed density values that the Ayalon data cover is larger; therefore, the larger number of clusters may be due to the need to capture the wider range of data. Furthermore, it is evident that the Ayalon data have a smaller variability. This may be due to the characteristics of the flow, but are more likely to be due to the larger aggregation of the data (to 5-min intervals, instead of 2-min intervals as in the Irvine case). In the case of the Irvine data, larger variability does not necessarily mean that more clusters can be statistically identifiable,



Context tree for Ayalon (8 clusters)

Context tree for Irvine (4 clusters)

Context tree for Ayalon (5 clusters)

Fig. 7. Context trees for variable length Markov chain models.

Table 1
Traffic state prediction results (left: Irvine, right: Ayalon).

Irvine, CA

predicted cluster

4 clusters - knn/vlmc
True cluster

	1	2	3	4
1	**171**		24	
2		**60**	17	6
3	15	17	**86**	
4		6		**17**

3 clusters - knn/vlmc
True cluster

	1	2	3
1	**208**	31	
2	17	**107**	8
3		8	**40**

4 clusters - nnet
True cluster

	1	2	3	4
1	**174**		17	
2		**62**	13	6
3	17	14	**89**	
4				**19**

3 clusters - nnet
True cluster

	1	2	3
1	**203**	19	
2	19	**117**	9
3		9	**43**

Ayalon, IL

predicted cluster

8 clusters – knn/vlmc
True cluster

	1	2	3	4	5	6	7	8
1	**25**	1	2					
2	1	**47**						
3	2		**48**	3				1
4			4	**71**	1	6	1	1
5					**12**		2	
6				6		**4**	1	1
7				1	1	2	**25**	2
8				3			2	**11**

5 clusters - knn/vlmc
True cluster

	1	2	3	4	5
1	**60**	2	2		1
2	2	**58**			
3	3		**83**	4	2
4			3	**16**	3
5			4	2	**42**

3 clusters - knn/vlmc
True cluster

	1	2	3
1	**113**	6	1
2	3	**111**	5
3		6	**42**

predicted cluster

8 clusters - nnet
True cluster

	1	2	3	4	5	6	7	8
1								
2	16	**59**	2					
3	2		**47**	2				2
4			4	**56**	2	2	4	1
5				1	**12**	1	1	
6				6		**7**		1
7								
8				4	1	4	5	**45**

5 clusters - nnet
True cluster

	1	2	3	4	5
1	**54**	4	5		
2	4	**58**			
3	5		**54**	5	5
4			6	**23**	3
5			4	4	**53**

3 clusters - nnet
True cluster

	1	2	3
1	**107**	4	
2	4	**93**	12
3		12	**55**

as there is more mass, and the underlying distributions have greater overlap. Ayalon, on the other hand, has a crisper diagram, so it might be easier to define (smaller) regions.

3.3.2. Speed prediction

Loess curves have been estimated for each cluster (and the entire sample) using the traffic data (speed, flow and density). Following the prediction of the state, it is possible to use the estimated loess curve for that cluster, along with the density and flow for that observation, to perform speed prediction. Table 2 summarizes the prediction results for the cases presented in Section 3.1. Overall, the results are encouraging, with prediction accuracy of about 3–4%, according to the RMSN and RMSPE metrics). This represents an improvement of about 50% from the typical speed density relationship for both data sets. The MPE measure is in general low. For example, MPE for Irvine do not indicate an advantage for the proposed approach, while all values for prediction are negative. However, MPE is already essentially 0 for the base case (no bias in the predictions), so there is no further advantage to be gained. The values may have a negative sign, but they are extremely small (essentially zero, if e.g. they are rounded to 2 decimal digits). Also, the values indicated by the Theil inequality coefficient components are considerably improved after the application of the complete framework and in any case they have very low values in absolute terms. In addition, the following observations can be made:

- Locally weighted regression applied to the entire dataset (i.e. without clustering) provides superior performance to the typical speed–density relationship. This is expected, as (i) it can integrate additional explanatory variables, and (ii) its functional form is less restricted (i.e. can better follow the data).
- Decreasing the number of clusters from 8 to 5 (in the Ayalon data set), in the application of the full-blown methodology, does not significantly affect the performance (in terms of accuracy in traffic state and local speed prediction). Further reduction (to three clusters for the Ayalon data set) provides a deterioration, but still much better performance than the typical speed–density relationship. Similarly, decreasing the number of clusters from 4 to 3 (in the Irvine data set) does not significantly affect the performance (in terms of accuracy).
- The proposed framework provides considerable benefits over the state-of-the-art (and often comparable with the full framework) also when a neural network is used in lieu of the clustering and classification algorithms.

One question that might arise from these results is whether there is any differentiation in performance of the proposed algorithms under different conditions. For example, does the proposed approach outperform the reference cases both under peak and off-peak conditions? To answer this question, the results of Table 2 are presented in Figs. 8 and 9 for all data (top), peak period data (middle) and off-peak (bottom) conditions. Fig. 8 presents the results for the Irvine network and Fig. 9 for the Ayalon dataset. The results for the two networks suggest that the performance of the proposed approach does not depend on the prevailing traffic conditions. Having said that, there seems to be some differentiation between the results for the two networks, with the ones for Irvine showing lower variability, while those for Ayalon show a higher variability. This is consistent with the observations made earlier, regarding the spectrum of data available for the Ayalon network, which is wider than that for the Irvine dataset.

Table 2
Summary results of speed prediction (top: Irvine, bottom: Ayalon).

	Irvine					
	I. Speed–density relationship	II. Loess – no clustering	III. Prediction (loess, clustering)			
			a. 4 clusters (knn/vlmc)	b. 3 clusters (knn/vlmc)	c. 4 clusters (nnet)	d. 3 clusters (nnet)
RMSN	0.0702	0.0293	0.0255	0.0251	0.0253	0.0255
RMSPE	0.0807	0.0381	0.0269	0.0271	0.0271	0.0277
MPE	0.0016	−0.0074	−0.0008	−0.0020	−0.0015	−0.0015
U	0.0346	0.0144	0.0126	0.0124	0.0124	0.0126
Um	0.0102	0.0244	0.0008	0.0043	0.0023	0.0024
Us	0.0751	0.1032	0.0018	0.0065	0.0033	0.0030

	Ayalon							
	I. Speed–density relationship	II. Loess – no clustering	III. Prediction (loess, clustering)					
			a. 8 clusters (knn/vlmc)	b. 5 clusters (knn/vlmc)	c. 3 clusters (knn/vlmc)	d. 8 clusters (nnet)	e. 5 clusters (nnet)	f. 3 clusters (nnet)
RMSN	0.0829	0.0490	0.0320	0.0420	0.0312	0.0508	0.0488	0.0477
RMSPE	0.0941	0.0532	0.0347	0.0459	0.0346	0.0530	0.0505	0.0494
MPE	0.0177	0.0032	0.0040	0.0044	0.0051	0.0000	0.0015	0.0016
U	0.0401	0.0238	0.0156	0.0204	0.0151	0.0247	0.0237	0.0232
Um	0.0166	0.0000	0.0036	0.0042	0.0109	0.0008	0.0000	0.0000
Us	0.0056	0.0049	0.0204	0.0013	0.0166	0.0002	0.0002	0.0000

The Theil Us coefficient shows some rather high values in some cases. For example, it is higher for the off-peak conditions. As discussed, Us measures how well the model is able to replicate the variability in the observed data. When the off-peak data are considered, while the observed data have some variability (due to variability in desired speeds for example), the predicted data correspond more closely to the free flow speed and therefore have a very low standard deviation, especially compared to the actual data. This is more of a problem with the basic approach, which uses the speed–density relationship of Eq. (8), and less so with the other methods.

4. Discussion

A methodology for the identification and short-term prediction of traffic state and local speed, designed to take advantage of the ever-increasing availability of traffic data through emerging sensors, has been presented. The methodology is a two-step approach, where in the first step an observation is assigned to a traffic state and then a state-specific function is used to estimate/predict the corresponding speed. An application of the methodology to short-term speed prediction in freeway datasets in Irvine, CA, and Tel Aviv, Israel, provides encouraging results. The two data sets have somewhat different properties (in terms of coverage of the range of possible traffic conditions, as well as variability), which helps identify how the proposed approaches perform under different conditions. The results also show that increasing the number of clusters (providing a finer description of traffic states) does not result in better performance, as it also increases misclassification errors and over-fitting. Instead, a small number of traffic states provide a sufficient description of traffic conditions.

In addition to further testing to validate the proposed methodology (e.g. in more networks, including urban arterials), a number of potential directions for further research include:

- Incorporation of additional explanatory variables to capture impact of geometric characteristics, downstream conditions, weather conditions, etc. The proposed methodology is well suited to include such variables. The enrichment of the data driven models with such additional information may enable them to capture traffic characteristics that cannot be otherwise modeled explicitly. Furthermore, the gradual enrichment of the "system knowledge" database (Fig. 1) may result in increasingly more advanced models that will evolve and adapt to changing traffic dynamics.
- Robustness of the methodology with respect to measurement and other errors that corrupt the data. In-depth analysis of alternative algorithms and specific model structures to be used within each component of the methodology.
- A number of parameters were estimated using strictly optimization criteria (usually of some statistical measure, such as BIC). It would be interesting to explore the impact of choosing different parameter values that resulted in a marginally lower value of the objective function, but might provide superior performance.

The main contributions of this paper include the development of a methodology for traffic state prediction, based on a set of flexible models, both in terms of functional specification and data to which they can be applied. As the proposed models are shown to outperform current state-of-the-art models, they could be valuable when integrated into existing traffic simulation models, resulting in more accurate traffic predictions. These predictions could then better support downstream applications, such as traffic guidance generation. The data-driven algorithms that are integrated in the presented framework are readily implemented in widely used statistical software and as such could be easily interfaced with traffic simulation models. Alternatively, to improve efficiency, these algorithms are well documented and therefore could also be implemented directly within the traffic simulation systems.

A number of applications can be explored. One of the promising applications of the presented data-driven models is their introduction into mesoscopic traffic simulation models (for example the ones used by state-of-the-art simulation-based DTA models, DynaMIT, Ben-Akiva et al., 2002, 2010, DYNASMART, Mahmassani, 2001, or RENAISSANCE, Wang et al., 2006a,b) and the assessment of their performance and impact in such an environment. For each time step the model would then use the density, flow and other explanatory variables at each link to estimate the speed of the impacted vehicles. In the data-sets available for this research, no additional data – besides traffic variables – were available, and therefore it was not possible to further demonstrate the benefits of the developed approach. The application of the proposed approach to richer data sets is expected to further demonstrate its contribution to the development of more accurate traffic state prediction models. This becomes particularly relevant in the context of emerging data collection possibilities and opportunistic data sets (a review of which can be found in Antoniou et al., 2011), which can be easily incorporated into the proposed model framework without requiring reformulations.

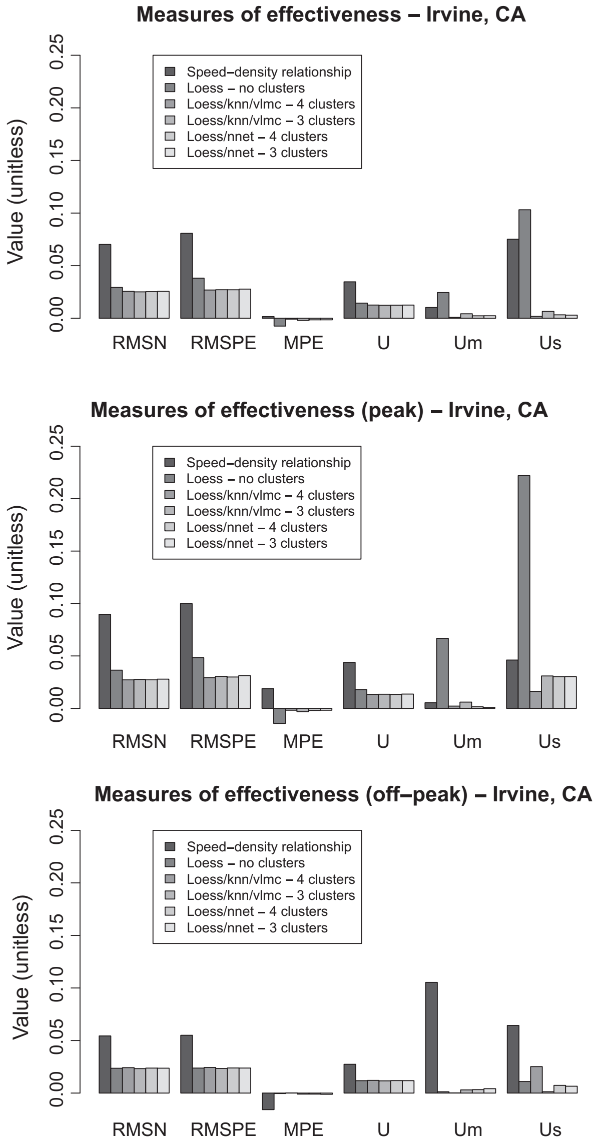


Fig. 8. Comparison of the measures of effectiveness for all scenarios for the Irvine network (top: entire period, middle: peak conditions, bottom: off-peak conditions).

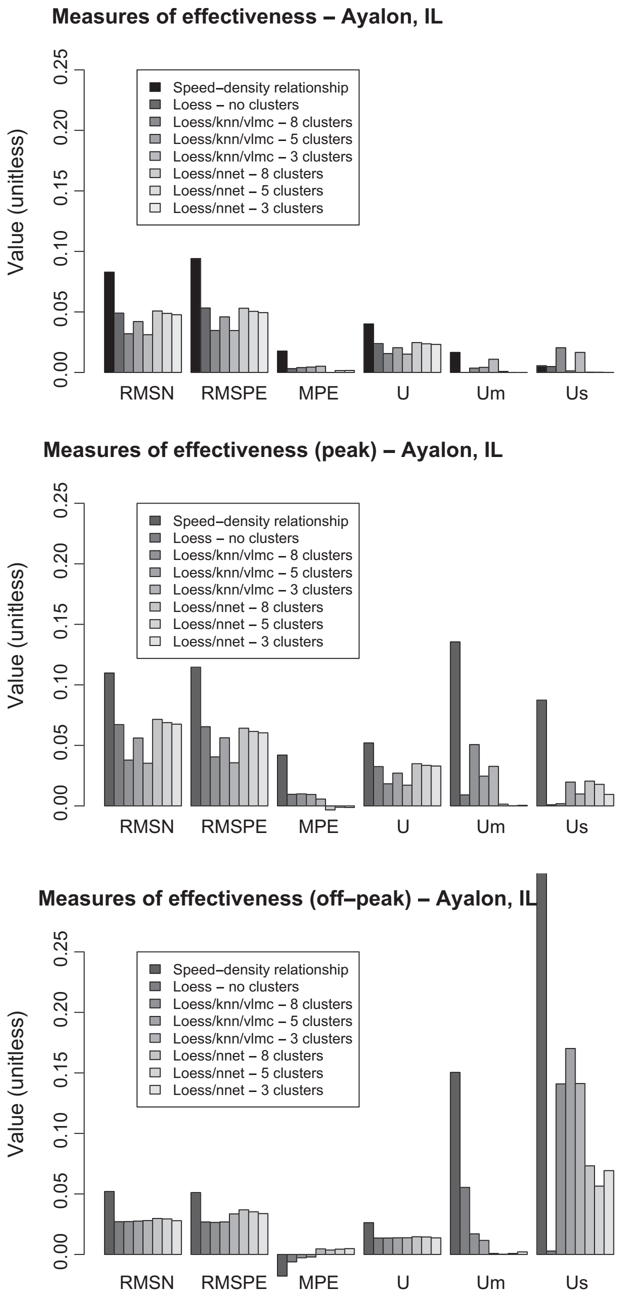


Fig. 9. Comparison of the measures of effectiveness for all scenarios for the Ayalon network (top: entire period, middle: peak conditions, bottom: off-peak conditions).

Other potential applications in the field of motorway surveillance and control, in addition to local prediction of speed, include automated incident detection and capacity estimation. Incident detection, or at least a warning that conditions are drastically different than the ones expected, can be achieved when the observed traffic state differs from the state that was predicted by the model. A deviation from this expectation may suggest that some special event has disrupted the normal flow of traffic and trigger an intelligent system or the traffic management center operator to react quickly.

Acknowledgements

The authors would like to thank Prof. Wilfred Recker from University of California, Irvine, and Dr. Tomer Toledo from Technion Institute of Technology, Israel, for providing the data used in this research.

References

Abaza, K.A., Ashur, S.A., Al-Khatib, I.A., 2004. Integrated pavement management system with a Markovian prediction model. Journal of Transportation Engineering 130 (1), 24–33.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19 (6), 716–723.

Antoniou, C., Koutsopoulos, H.N., 2006a. Estimation of traffic dynamics models with machine learning methods. Transportation Research Record 1965, 103–111 (Washington, DC).

Antoniou, C., Koutsopoulos, H.N., 2006b. A comparison of machine learning methods for speed estimation. In: Proceedings of the 11th IFAC Symposium on Control in Transportation Systems, Delft, The Netherlands, August 29–31, 2006.

Antoniou, C., Balakrishna, R., Koutsopoulos, H.N., 2011. A synthesis of emerging data collection technologies and their impact on traffic management applications. European Transport Research Review 3 (3), 139–148.

Atkenson, C.G., Moore, A., Schaal, S., 1997. Locally weighted learning. AI Review 11, 11–73.

Azimi, M., Zhang, Y., 2010. Categorizing freeway flow conditions using clustering methods. In: Proceedings of the 89th Annual Meeting of the Transportation Research Board, Washington, DC.

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H.N., Mishalani, R., 2002. Real-time simulation of traffic demand–supply interactions within DynaMIT. In: Gendreau, M., Marcotte, P. (Eds.), Transportation and Network Analysis: Current Trends. Kluwer Academic Publishers, Boston/Dordrecht/London, pp. 19–36 (Miscellenea in Honor of Michael Florian).

Ben-Akiva, M., Koutsopoulos, H.N., Antoniou, C., Balakrishna, R., 2010. Traffic simulation with DynaMIT. In: Barcelo, J. (Ed.), Fundamentals of Traffic Simulation. Springer.

Binder, D.A., 1978. Bayesian cluster analysis. Biometrika 65, 31–38.

Bock, H.H., 1998a. Probabilistic approaches in cluster analysis. Bulletin of the International Statistical Institute 57, 603–606.

Bock, H.H., 1998b. Probabilistic aspects in classification. In: Hayashi, C., Yajima, K., Bock, H.H., Oshumi, N., Tanaka, Y., Baba, Y. (Eds.), Data Science, Classification and Related Methods. Springer-Verlag, New York, pp. 3–21.

Burghout, W., Koutsopoulos, H., Andreasson, I., 2005. Hybrid mesoscopic–microscopic traffic simulation. Transportation Research Record 1934, 218–225.

Cheeseman, P., Stutz, J., 1995. Bayesian classification (AutoClass): theory and results. In: Fayyad, U., Piatesky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.), Advances in Knowledge Discovery and Data Mining. AAAI Press, pp. 153–180.

Cleveland, W.S., 1978. Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association 74, 829–836.

Cleveland, W.S., Devlin, S.J., 1988. Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association 83, 596–610.

Cleveland, W.S., Devlin, S.J., Grosse, E., 1988. Regression by local fitting: methods, properties and computational algorithms. Journal of Econometrics 37, 87–114.

Day, N.E., 1969. Estimating the components of a mixture of normal distributions. Biometrika 56, 463–474.

Drake, J., Schofer, J., May, A.D., 1967. A statistical analysis of speed density hypotheses. In: Third International Symposium on the Theory of Traffic Flow Proceedings. Elsevier North Holland, Inc, New York.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Dunne, S., Ghosh, B., 2012. Regime-based short-term multivariate traffic condition forecasting algorithm. Journal of Transportation Engineering 138 (4), 455–466.

Edie, L.C., 1961. Following and steady-state theory for non-congested traffic. Operations Research 9, 66–76.

Edwards, A.W.F., Cavalli-Sforza, L.L., 1965. A method for cluster analysis. Biometrics 21, 362–375.

Einbeck, J., Tutz, G., 2004. Modelling beyond Regression Functions: An Application of Multimodal Regression to Speed-Flow Data. SFB Discussion Paper 395.

El Faouzi, N.-E., 2004. Data-driven aggregative schemes for multisource estimation fusion: a road travel time application. In: Dasarathy, Belur V. (Ed.), Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2004, Proceedings of SPIE, vol. 5434. SPIE, Bellingham, WA, pp. 351–359.

El Faouzi, N.-E., Lefevre, E., 2006. Classifiers and distance-based evidential fusion for road travel time estimation. In: Dasarathy, Belur V. (Ed.), Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006. Proceedings of SPIE, vol. 6242.

El Faouzi, N.-E., Klein, L.A., Mouzon, O.D., 2009. Improving travel time estimates from inductive loop and toll collection data with Dempster–Shafer data fusion. Transportation Research Record 2129, 73–80.

Fraley, C., Raftery, A.E., 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer Journal 41, 578–588.

Fraley, C., Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97 (458), 611–631.

Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. Journal of the American Statistical Association 62, 1159–1178.

Geroliminis, N., Skabardonis, A., 2005. Prediction of arrival profiles and queue lengths along signalized arterials using a Markov decision process. Transportation Research Record 1934, 116–124.

Hall, F.L., 1997. Traffic stream characteristics. In: Gartner, N.H., Messer, C.J., Rathi, A.K. (Ed.), Monograph of Traffic Flow Theory. <http://www.tfhrc.gov/its/tft/tft.htm> (accessed 05.11.06).

Haykin, S., 1999. Neural Networks, second ed. Prentice Hall.

Hollander, Y., Liu, R., 2008. The principles of calibrating traffic microsimulation models. Transportation 35, 347–362.

Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: differences, similarities and some insights. Transportation Research Part C 19, 387–399.

Kerner, B.S., 2004. The Physics of Traffic. Springer, Heidelberg.

Kerner, B.S., Klenov, S.L., 2002. A microscopic model for phase transitions in traffic flow. Journal of Physics A 35 (3), 31–43.

Liu, H., Van Zuylen, H., Van Lint, H., Salomons, M., 2006. Predicting urban arterial travel time with state-space neural networks and Kalman filters. Transportation Research Record, 99–108.

Maechler, M., Buehlmann, P., 2004. Variable length Markov chains: methodology, computing, and software. Journal of Computational and Graphical Statistics 2, 435–455.

Mahmassani, H.S., 2001. Dynamic network traffic assignment and simulation methodology for advanced system management applications. Networks and Spatial Economics 1 (3), 267–292.

Markov, A.A., 1971. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. Reprinted in Appendix B of: R. Howard. Dynamic Probabilistic Systems, volume 1: Markov Chains. John Wiley and Sons.

May, A., 1990. Traffic Flow Fundamentals. Prentice Hall, New Jersey.

May, A.D., Keller, H.E.M., 1967. Non-integer Car-Following Models. Highway Research Board, Record 199, Washington, DC, pp. 19–32.

McLachlan, G.J., Basford, K.E., 1988. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York.

Mitchell, T., 1997. Machine Learning. McGraw Hill.

Murtagh, F., Raftery, A.E., 1984. Fitting straight lines to point patterns. Pattern Recognition 17, 479–483.

Ortiz-Garcia, J.J., Costello, S.B., Snaith, M.S., 2006. Derivation of transition probability matrices for pavement deterioration modeling. Journal of Transportation Engineering 132 (2), 141–161.

Pindyck, R.S., Rubinfeld, D.L., 1997. Econometric Models and Economic Forecasts, fourth ed. Irwin McGraw-Hill, Boston, MA.

R Development Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org> (accessed 06.01.12).

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge.

Rissanen, J., 1983. A universal data compression system. IEEE Transactions on Information Theory IT-29, 656–664.

Scherer, W.T., Glagola, D.M., 1994. Markovian models for bridge maintenance management. Journal of Transportation Engineering 120 (1), 37–51.

Schoenhof, M., Helbing, D., 2009. Criticism of three-phase traffic theory. Transportation Research Part B 43, 784–797.

Schwarz, G., 1978. Estimating the dimension of a model. Annals of Statistics 6 (2), 461–464.

Scott, A.J., Symons, M.J., 1971. Clustering methods based on likelihood ratio criteria. Biometrics 27, 387–397.

Stamoulakatos, T.S., Sykas, E.D., 2007. Hidden Markov modeling and macroscopic traffic filtering supporting location-based services. Wireless Communications and Mobile Computing 7 (4), 415–429.

Stathopoulos, A., Karlaftis, M.G., 2003. A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies 11 (2), 121–135.

Sun, H., Liu, H., Xiao, H., He, R., Ran, B., 2003. Use of local linear regression model for short term traffic forecasting, Transportation Research Record 1836 (Washington, DC).

Sun, L., Zhou, J., 2005. Developing multi-regime speed–density relationships using cluster analysis. Transportation Research Record: Journal of the Transportation Research Board 1934, 64–71 (DC).

Theil, H., 1961. Economic Forecasts and Policy. North-Holland, Amsterdam, The Netherlands.

Toledo, T., Koutsopoulos, H.N., 2004. Statistical validation of traffic simulation models. Transportation Research Record 1876, 142–150.

Toledo, T., Koutsopoulos, H.N., Ahmed, K., 2007. Estimation of vehicle trajectories with locally weighted regression. Transportation Research Record: Journal of the Transportation Research Board 1999, 161–169.

van Lint, J.W.C., 2008. Online learning solutions for freeway travel time prediction. IEEE Transactions on Intelligent Transportation Systems 9 (1), 38–47.

van Lint, J.W.C., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transportation Research Part C: Emerging Technologies 13, 347–369.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer.

Vapnik, V., 1998. Statistical Learning Theory. Wiley-Interscience, New York.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S, fourth ed. Springer.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transportation Research Part C 13 (3), 211–234.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2008. Temporal evolution of short-term urban traffic flow: a nonlinear dynamics approach. Computer-Aided Civil and Infrastructure Engineering 23, 536–548.

Wang, Y., Papageorgiou, M., Messmer, A., 2006a. A real-time freeway network traffic surveillance tool. IEEE Transactions on Control Systems Technology 14 (2006), 18–32.

Wang, Y., Papageorgiou, M., Messmer, A., 2006b. RENAISSANCE – a unified macroscopic model-based approach to real-time freeway network traffic surveillance. Transportation Research Part C: Emerging Technologies 14, 190–212.

Wang, L., Rong, J., Liu, X., 2005. The classification of car-following behavior in urban expressway based on fuzzy clustering analysis. In: Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington, DC.

Wolfe, J.H., 1970. Pattern clustering by multivariate mixture analysis. Multivariate Behavioral Research 5, 329–350.

Yeon, J., Elefteriadou, L., Lawphongpanich, S., 2008. Travel time estimation on a freeway using discrete time Markov chains. Transportation Research Part B 42 (2008), 325–338.