

Cooperative Kernel-Based Forecasting in Decentralized Multi-Agent Systems for Urban Traffic Networks

Jelena Fiosina and Maksims Fiosins¹

Abstract. The distributed and often decentralised nature of complex stochastic traffic systems having a large amount of distributed data can be represented well by multi-agent architecture. Traditional centralized data mining methods are often very expensive or not feasible because of transmission limitations that lead to the need of the development of distributed or even decentralized data mining methods including distributed parameter estimation and forecasting. We consider a system, where drivers are modelled as autonomous agents. We assume that agents possess an intellectual module for data processing and decision making. All such devices use a local kernel-based regression model for travel time estimation and prediction. In this framework, the vehicles in a traffic network collaborate to optimally fit a prediction function to each of their local measurements. Rather than transmitting all data to one another or a central node in a centralized approach, the vehicles carry out a part of the computations locally by transmitting only limited amount of data. This limited transmission helps the agent that experiences difficulties with its current predictions. We demonstrate the efficiency of our approach with case studies with the analysis of real data from the urban traffic domain.

1 INTRODUCTION

Multi-agent systems (MAS) often deal with complex applications, such as sensors, traffic, or logistics networks, and they offer a suitable architecture for distributed problem solving. In such applications, the individual and collective behaviours of the agents depend on the observed data from distributed sources. In a typical distributed environment, analysing distributed data is a non-trivial problem because of several constraints such as limited bandwidth (in wireless networks), privacy-sensitive data, distributed computing nodes, etc. The distributed data processing and mining (DDPM) field deals with these challenges by analysing distributed data and offers many algorithmic solutions for data analysis, processing, and mining using different tools in a fundamentally distributed manner that pays careful attention to the resource constraints [3].

Traditional centralized data processing and mining typically requires central aggregation of distributed data, which may not always be feasible because of the limited network bandwidth, security concerns, scalability problems, and other practical issues. DDPM carries out communication and computation by analyzing data in a distributed fashion [10]. The DDPM technology offers more efficient solutions in such applications.

In this study we focus on the urban traffic domain, where many traffic characteristics such as travel time, travel speed, congestion

probability, etc. can be evaluated by autonomous agents-vehicles in a distributed manner. In this paper, we present a general distributed regression forecasting algorithm and illustrate its efficiency in forecasting travel time.

Numerous data processing and mining techniques were suggested for forecasting travel time in a centralized and distributed manner. Statistical methods, such as regression and time series, and artificial intelligence methods, such as neural networks, are successfully implemented for similar problems. However, travel time is affected by a range of different factors. Thus, accurate prediction of travel time is difficult and needs considerable amount of traffic data. Understanding the factors affecting travel time is essential for improving prediction accuracy [13].

We focus on non-parametric, computationally intensive estimation, i.e. Kernel-based estimation, which is a promising technique for solving many statistical problems, including parameter estimation. In our paper, we suggest a general distributed kernel-based algorithm and use it for forecasting travel time using real-world data from southern Hanover. We assume that each agent autonomously estimates its kernel-based regression function, whose additive nature fits very well with streaming real-time data. When an agent is not able to estimate the travel time because of lack of data, i.e., when it has no data near the point of interest (because the kernel-based estimation uses approximations), it cooperates with other agents. An algorithm for multi-agent cooperative learning based on transmission of the required data as a reply to the request of the agent experiencing difficulties was suggested. After obtaining the necessary data from other agents, the agent can forecast travel-time autonomously.

The travel-time, estimated in the DDPM stage, can serve as an input for the next stage of distributed decision making of the intelligent agents [5].

This study contributes in the following ways: It suggests (a) a decentralized kernel-based regression forecasting approach, (b) a regression model with a structure that facilitates travel-time forecasting; and it improves the application efficiency of the proposed approach for the current real-world urban traffic data.

The remainder of this paper is organized as follows. Section 2 describes the related previous work in the DDPM field for MAS, kernel density estimation, and travel-time prediction. Section 3 describes the current problem more formally. Section 4 presents the multivariate kernel-based regression model adopted for streaming data. Section 5 describes the suggested cooperative learning algorithm for optimal prediction in a distributed MAS architecture. Section 6 presents a case study and the final section contains the conclusions.

¹ Institute of Informatics, Clausthal University of Technology, Germany
email: {jelena.fiosina, maksims.fiosins}@gmail.com

2 RELATED PREVIOUS WORK

2.1 Distributed Data Mining in Multi-agent Systems

A strong motivation for implementing DDPM for MAS is given by Da Silva et al. in [3], where authors argue that DDPM in MAS deals with pro-active and autonomous agents that perceive their environment, dynamically reason out actions on the basis of the environment, and interact with each other. In many complex domains, the knowledge of agents is a result of the outcome of empirical data analysis in addition to the pre-existing domain knowledge. DDPM of agents often involves detecting hidden patterns, constructing predictive and clustering models, identifying outliers, etc. In MAS, this knowledge is usually collective. This collective 'intelligence' of MAS must be developed by distributed domain knowledge and analysis of the distributed data observed by different agents. Such distributed data analysis may be a non-trivial problem when the underlying task is not completely decomposable and computational resources are constrained by several factors such as limited power supply, poor bandwidth connection, privacy-sensitive multi-party data.

Klusch et al. [11] concludes that autonomous data mining agents, as a special type of information agents, may perform various kinds of mining operations on behalf of their user(s) or in collaboration with other agents. Systems of cooperative information agents for data mining in distributed, heterogeneous, or homogeneous, and massive data environments appear to be quite a natural progression for the current systems to be realized in the near future.

A common feature of all approaches is that they aim at integrating the knowledge that is extracted from data at different geographically distributed network sites with minimum network communication and maximum local computations. Local computation is carried out on each site, and either a central site communicates with each distributed site to compute the global models or a peer-to-peer architecture is used. In the case of the peer-to-peer architecture, individual nodes might communicate with a resource-rich centralized node, but they perform most tasks by communicating with neighbouring nodes through message passing over an asynchronous network [3].

A distributed system should have the following features for the efficient implementation of DDPM: the system consists of multiple independent data sources, which communicate only through message passing; communication between peers is expensive; peers have resource constraints (e. g. battery power) and privacy concerns [3].

Typically, communication involves bottlenecks. Since communication is assumed to be carried out exclusively by message passing, the primary goal of several DDPM methods, as mentioned in the literature, is to minimize the number of messages sent. Building a monolithic database in order to perform non-distributed data processing and mining may be infeasible or simply impossible in many applications. The costs of transferring large blocks of data may be very expensive and result in very inefficient implementations [10].

Moreover, sensors must process continuous (possibly fast) streams of data. The resource-constrained distributed environments of sensor networks and the need for a collaborative approach to solve many problems in this domain make MAS architecture an ideal candidate for application development.

In our study we deal with homogeneous data. However, a promising approach to agent-based parameter estimation for partially heterogeneous data in sensor networks was suggested in [7]. Another decentralized approach for homogeneous data was suggested in [18] to estimate the parameters of a wireless network by using a parametric linear model and stochastic approximations.

2.2 Travel Time Prediction Models

Continuous traffic jams indicate that the maximum capacity of a road network is met or even exceeded. In such a situation, the modelling and forecasting of traffic flow is one of the important techniques that need to be developed [1]. Nowadays, knowledge about travel time plays an important role in transportation and logistics, and it can be applied in various fields and purposes. From travellers' viewpoints, the knowledge about travel time helps to reduce the travel time and improves reliability through better selection of travel routes. In logistics, accurate travel time estimation could help reduce transport delivery costs and increase the service quality of commercial delivery by delivering goods within the required time window by avoiding congested sections. For traffic managers, travel time is an important index for traffic system operation efficiency [13].

There are several studies in which a centralized approach is used to predict the travel time. The approach was used in various intelligent transport systems, such as in-vehicle route guidance and advanced traffic management systems. A good overview is given in [13]. To make the approach effective, agents should cooperate with each other to achieve their common goal via the so called gossiping scenarios. The estimation of the actual travelling time using vehicle-to-vehicle communication without MAS architecture was described in [14].

On the other hand, a multi-agent architecture is better suited for distributed traffic networks, which are complex stochastic systems. Further, by using centralized approaches the system cannot adapt quickly to situations in real time, and it is very difficult to transmit a large amount of information over the network. In centralized approaches, it is difficult or simply physically impossible to store and process large data sets in one location. In addition, it is known from practice that the most drivers rely mostly on their own experience; they use their historical data to forecast the travel time [5]. Thus, decentralized multi-agent systems are fundamentally important for the representation of these networks [1]. We model our system with autonomous agents to allow vehicles to make decisions autonomously using not only the centrally processed available information, but also their historical data.

Traffic information generally goes through the following three stages: data collection and cleansing, data fusion and integration, and data distribution [12]. The system presented in [12] consists of three components, namely a Smart Traffic Agent, the Real-time Traffic Information Exchange Protocol and a centralized Traffic Information Centre that acts as the backend. A similar architecture is used in this study, but the prediction module, incorporated into Start Traffic Agent (vehicle agent), is different. In our study we do not focus on the transmission protocol describing only the information, which should be sent from one node to another, without the descriptions of protocol packets. The centralized Traffic Information Centre in our model is used only for storing system information.

The decentralized MAS approach for urban traffic network was considered in [2] also, where the authors forecast the traversal time for each link of the network separately. Two types of agents were used for vehicles and links, and a neural network was used as the prediction model.

For travel time forecasting different regression models can be applied. Linear multivariate regression model for decentralized urban traffic network was proposed in [4]. This regression model is well-studied, is parametric and allows estimation of each variables contribution. However is not sufficiently effective in the case of non-linear systems. Alternatively non-parametric kernel-based regression models can be applied. These models can be effectively used for any types

of systems, however are relatively new and not well-studied yet. The efficiency of non-parametric kernel-based regression approach for traffic flow forecasting in comparison to parametric approach was made in [17]. In this study we apply non-parametric kernel regression for the similar traffic system as in [4] in order to increase the prediction quality.

2.3 Kernel Density Estimation

Kernel density estimation is a non-parametric approach for estimating the probability density function of a random variable. Kernel density estimation is a fundamental data-smoothing technique where inferences about the population are made on the basis of a finite data sample. A kernel is a weighting function used in non-parametric estimation techniques.

Let X_1, X_2, \dots, X_n be an iid sample drawn from some distribution with an unknown density f . We attempt to estimate the shape of f , whose kernel density estimator is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right), \quad (1)$$

where kernel $K(\bullet)$ is a non-negative real-valued integrable function satisfying the following two requirements: $\int_{-\infty}^{\infty} K(u)du = 1$ and $K(-u) = K(u)$ for all values of u ; $h > 0$ is a smoothing parameter called bandwidth. The first requirement to $K(\bullet)$ ensures that the kernel density estimator is a probability density function. The second requirement to $K(\bullet)$ ensures that the average of the corresponding distribution is equal to that of the sample used [8]. Different kernel functions are available: Uniform, Epanechnikov, Gaussian, etc. They differ in the manner in which they take into account the vicinity observations to estimate the function from the given variables.

A very suitable property of the kernel function is its additive nature. This property makes the kernel function easy to use for streaming and distributed data [8], [3], [7]. In [11], the distributed kernel-based clustering algorithm was suggested on the basis of the same property. In this study, kernel density is used for kernel regression to estimate the conditional expectation of a random variable.

3 PROBLEM FORMULATION

We consider a traffic network with several vehicles, represented as autonomous agents, which predict their travel time on the basis of their current observations and history. Each agent estimates locally the parameters of the same traffic network. In order to make a forecast, each agent constructs a regression model, which explains the manner in which different explanatory variables (factors) influence the travel time. A detailed overview of such factors is provided in [13]. The following information is important for predicting the travel time [15]: average speed before the current segment, number of stops, number of left turns, number of traffic light, average travel time estimated by Traffic Management Centres (TMC). We should also take into account the possibility of an accident, network overload (rush hour) and the weather conditions.

Let us consider a vehicle, whose goal is to drive through the definite road segment under specific environment conditions (day, time, city district, weather, etc.). Let us suppose that it has no or little experience of driving in such conditions. For accurate travel time estimation, it contacts other traffic participants, which share their experience in the requested point.

In this step, the agent that experiences difficulties with a forecast sends its requested data point to other traffic participants in the transmission radius. The other agents try to make a forecast themselves. In the case of a successful forecast, the agents share their experience by sending their observations that are nearest to the requested point. After receiving the data from the other agents, the agent combines the obtained results, increases its experience, and makes a forecast autonomously.

In short, decentralized travel-time prediction consists of three steps: 1) local prediction; 2) in the case of unsuccessful prediction: selection of agents for experience sharing and sending them the requested data point; 3) aggregation of the answers and prediction.

4 LOCAL PARAMETER ESTIMATION

The non-parametric approach to estimating a regression curve has four main purposes. First, it provides a versatile method for exploring a general relationship between two variables. Second, it can predict observations yet to be made without reference to a fixed parametric model. Third, it provides a tool for finding spurious observations by studying the influence of isolated points. Fourth, it constitutes the flexible method of substitution or interpolating between adjacent \mathbf{X} -values for missing values [8].

Let us consider a non-parametric regression model [9] with a dependent variable Y and a vector of d regressors \mathbf{X}

$$Y = m(\mathbf{x}) + \epsilon, \quad (2)$$

where ϵ is the disturbance term such that $E(\epsilon|\mathbf{X} = \mathbf{x}) = 0$ and $Var(\epsilon|\mathbf{X} = \mathbf{x}) = \sigma^2(\mathbf{x})$, and $m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$. Further, let $(\mathbf{X}_i, Y_i)_{i=1}^n$ be the observations sampled from the distribution of (\mathbf{X}, Y) . Then the Nadaraya-Watson kernel estimator is

$$m_n(\mathbf{x}) = \frac{\sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right)} = \frac{p_n(\mathbf{x})}{q_n(\mathbf{x})}, \quad (3)$$

where $K(\bullet)$ is the kernel function of R^d and h is the bandwidth. Kernel functions satisfy the restrictions from (1). In our case we have a multi-dimensional kernel function $K(u) = K(u_1, u_2, \dots, u_d)$, that can be easily presented with univariate kernel functions as: $K(u) = K(u_1) \cdot K(u_2) \cdot \dots \cdot K(u_d)$. We used the Gaussian kernel in our experiments.

Network packets are streaming data. Standard statistical and data mining methods deal with a fixed dataset. There is a fixed size n for dataset and algorithms are chosen as a function of n . In streaming data there is no n : data are continually captured and must be processed as they arrive. It is important to develop algorithms that work with non-stationary datasets, handle the streaming data directly, and update their models on the fly [6].

This requires recursive windowing. The kernel density estimator has a simple recursive windowing method that allows the recursive estimation using the kernel density estimator:

$$m_n(\mathbf{x}) = \frac{p_n(\mathbf{x})}{q_n(\mathbf{x})} = \frac{p_{n-1}(\mathbf{x}) + K\left(\frac{\mathbf{x} - \mathbf{X}_n}{h}\right) Y_n}{q_{n-1}(\mathbf{x}) + K\left(\frac{\mathbf{x} - \mathbf{X}_n}{h}\right)}. \quad (4)$$

5 DECENTRALISED MULTI-AGENT COOPERATIVE LEARNING ALGORITHM

In this section, we describe the cooperation for sharing the prediction experience among the agents in a network. While working with

streaming data, one should take into account two main facts. The nodes should coordinate their prediction experience over some previous sampling period and adapt quickly to the changes in the streaming data, without waiting for the next coordination action.

Let us first discuss the cooperation technique. We introduce the following definitions.

Let $\mathbf{L} = \{L^j \mid 1 \leq j \leq p\}$ be a group of p agents. Each agent $L^j \in \mathbf{L}$ has a local dataset $D^j = \{(\mathbf{X}_c^j, Y_c^j) \mid c = 1, \dots, N^j\}$, where \mathbf{X}_c^j is a d -dimensional vector. In order to underline the dependence of the prediction function (3) from the local dataset of agent L^j , we denote the prediction function by $m[D^j](\mathbf{x})$.

Consider a case when some agent L^i is not able to forecast for some d -dimensional future data point \mathbf{X}_{new}^i because it does not have sufficient data in the neighbourhood of \mathbf{X}_{new}^i . In this case, it sends a request to other traffic participants in its transmission radius by sending the data point \mathbf{X}_{new}^i to them. Each agent L^j that has received the request tries to predict $m[D^j](\mathbf{X}_{new}^i)$. If it is successful, it replies to agent L^i by sending its best data representatives $\hat{D}^{(j,i)}$ from the neighbourhood of the requested point \mathbf{X}_{new}^i . Let us define $G^i \subset L$, a group of agents, which are able to reply to agent L^i by sending the requested data.

To select the best data representatives, each agent L^j makes a ranking among its dataset D^j . It can be seen from (3) that each Y_c^j is taken with the weight w_c^j with respect to \mathbf{X}_{new}^i , where

$$w_c^j = \frac{K\left(\frac{\mathbf{x}_{new}^i - \mathbf{x}_c^j}{h}\right)}{\sum_{l=1}^n K\left(\frac{\mathbf{x}_{new}^i - \mathbf{x}_l^j}{h}\right)}.$$

The observations with maximum weights w_c^j are the best candidates for sharing the experience.

All the data $\hat{D}^{(j,i)}$, $L^j \in G^i$ received by agent L^i should be verified, and duplicated data should be removed. We denote the new dataset of agent L^i as $D_{new}^i = \bigcup_{L^j \in G^i} \hat{D}^{(j,i)}$. Then, the kernel function of agent L^i is updated taking into account the additive nature of this function:

$$m[D_{new}^i](\mathbf{x}) = \sum_{L^j \in G^i} m[\hat{D}^{(j,i)}](\mathbf{x}) + m[D^i](\mathbf{x}). \quad (5)$$

Finally, agent L^i can autonomously make its forecast as $m[D_{new}^i](\mathbf{X}_{new}^i)$ for \mathbf{X}_{new}^i .

6 CASE STUDIES

We simulated a traffic network in the southern part of Hanover (Germany). The network contains three parallel and five perpendicular streets, creating fifteen intersections with a flow of approximately 5000 vehicles per hour.

The vehicles solve a travel time prediction problem. They receive information about the centrally estimated system variables (such as average speed, number of stops, congestion level, etc.) for this city district from TMC, combine it with their historical information, and make adjustments according to the information of other participants using the presented cooperative-learning algorithm. In this case, the regression analysis is an essential part of the local time prediction process. We consider the local kernel based regression model (3) and implement the cooperative learning algorithm (5). The factors are listed in Table 1. The selection and significance of these variables was considered in [4].

We simulated 20 agents having their initial experience represented by a dataset of size 20 till each agent made 100 predictions, thus

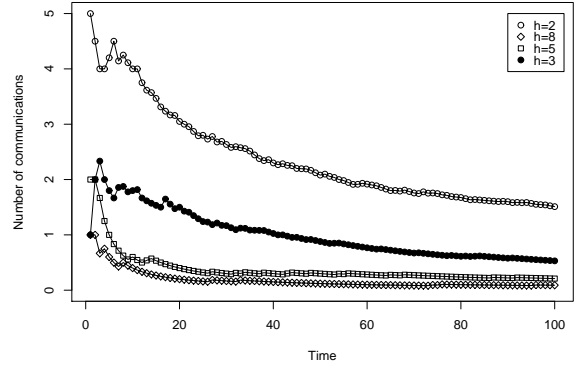


Figure 1. Average number of communications over time for different h

making their common experience equal to 2400. We assumed the maximal number of transmitted observations from one agent equals 2.

Table 1. System factors influencing the travel time

Variable	Description
Y	travel time (min);
X_1	length of the route (km);
X_2	average speed in the system (km/h);
X_3	average number of stops in the system (units/min);
X_4	congestion level of the flow (Veh/h);
X_5	number of traffic lights in the route (units);
X_6	number of left turns in the route (units).

During the simulation, to predict more accurately, the agents used the presented cooperative learning algorithm that supported the communication between agents with the objective of improving the prediction quality. The necessary number of communications depends on the value of the smoothing parameter h . The average number of necessary communications is given in Figure 1. We can see the manner in which the number of communications decreased with the learning time. We varied h and obtained the relation between the communication numbers and h as a curve. The prediction ability of one of the agents is presented at Figure 2. Here, we can also see the relative prediction error, which decreases with time. The predictions that used communication between agents are denoted by solid triangles, and the number of such predictions also decreases with the time.

The goodness of fit of the system was estimated using a cross-validation technique. We assume that each agent has its own training set, but it uses all observations of other agents as a test set, so we use 20-fold cross-validation. To estimate the goodness of fit, we used analysis of variance and generalized coefficient of determination R^2 that provides an accurate measure of the effectiveness of the prediction of future outcomes by using the non-parametric model [16]. The calculated R^2 values and the corresponding number of the observations that were not predicted (because cooperation during testing was not allowed) depending on h are listed in Table 2. From Figure 3 we can also see how R^2 is distributed among the system agents. The results suggest that we should find some trade-off between system

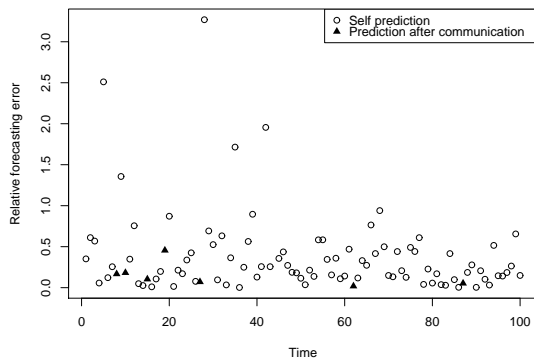


Figure 2. Relative prediction error and communication frequency of a single agent over time

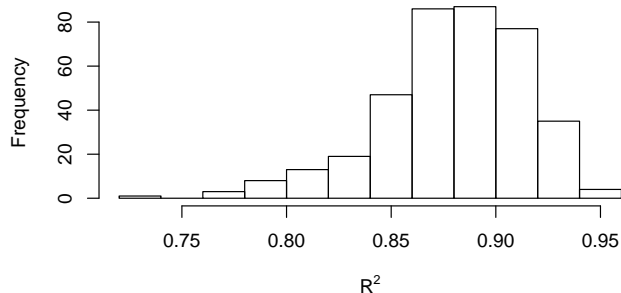


Figure 3. Distribution of goodness of fit measure R^2 using cross-validation for the whole system, $h = 2$

accuracy (presented by R^2) and the number of necessary communications (presented by the percentage of not predicted observations), which depend on h . The point of trade-off should depend on the communication and accuracy costs.

Table 2. R^2 goodness of fit measure using cross-validation for the whole system for different h

Characteristic of System	$h=2$	$h=3$	$h=5$	$h=8$
System average R^2	0.86	0.84	0.83	0.82
Average % of not predicted observations	3	1	0.5	0.1

A linear regression model [4] applied to the same data gives lower average goodness of fit $R^2=0.77$, however predictions can be calculated for all data points.

7 CONCLUSIONS

In this study, the problem of travel-time prediction was considered. Multi-agent architecture with autonomous agents was implemented for this purpose. Distributed parameter estimation and cooperative

learning algorithms were presented, using the non-parametric kernel-based regression model. We demonstrated the efficiency of the suggested approach through simulation with real data from southern Hanover. The experimental results show the high efficiency of the proposed approach. In future we are going to develop a combined approach that allows agent to choose between parametric and non-parametric estimator for more accurate prediction.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. PIEF-GA-2010-274881. We thank Prof. J. P. Müller for useful discussions during this paper preparation.

REFERENCES

- [1] A.L. Bazzan, J. Wahle, and F. Kluegl, 'Agents in traffic modelling - from reactive to social behaviour', *Advances in Artificial Intelligence (LNAI)*, **1701**, 509–523, (1999).
- [2] R. Claes and T. Holvoet, 'Ad hoc link traversal time prediction', in *Proc. of the 14th Int. IEEE Conf. on Intelligent Transportation Systems*, (October 2011).
- [3] J.C. da Silva, C. Giannella, R. Bhargava, H. Kargupta, and M. Klusch, 'Distributed data mining and agents', *Eng. Appl. Artif. Intell.*, **18**(7), 791–801, (2005).
- [4] J. Fiosina, 'Decentralised regression model for intelligent forecasting in multi-agent traffic networks', *Distributed Computing and Artificial Intelligence (Advances in Intelligent and Soft Computing)*, **151**, 255–263, (2012).
- [5] M. Fiosins, J. Fiosina, J. Müller, and J. Görmer, 'Agent-based integrated decision making for autonomous vehicles in urban traffic', *Advances on Practical Applications of Agents and Multiagent Systems (Advances in Intelligent and Soft Computing)*, **88**, 173–178, (2011).
- [6] *Handbook of Computational Statistics: Concepts and Methods*, eds., J.E. Gentle, W. Härdle, and Y. Mori, Springer, Berlin/Heidelberg, 2004.
- [7] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, 'Distributed regression: an efficient framework for modeling sensor network data', in *Proc. of the 3rd Int. Sym. on Information Processing in Sensor Networks*, Berkeley, (2004).
- [8] W. Härdle, *Applied Nonparametric Regression*, Cambridge University Press, Cambridge, 2002.
- [9] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, *Nonparametric and Semiparametric Models*, Springer, Berlin/Heidelberg, 2004.
- [10] *Advances in Distributed and Parallel Knowledge Discovery*, eds., H. Kargupta and P. Chan, AAAI Press The MIT Press/Academic Press, Melno Park and Cambridge and London, 2000.
- [11] M. Klusch, S. Lodi, and G. Moro, 'Agent-based distributed data mining: The kdec scheme', in *Proc. of Int. Conf. on Intelligent Information Agents - The AgentLink Perspective*, volume 2586 of *LNCS*, pp. 104–122. Springer, (2003).
- [12] W. Lee, S. Tseng, and W. Shieh, 'Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system', *Information Sciences*, **180**, 62–70, (2010).
- [13] H. Lin, R. Zito, and M.A.P. Taylor, 'A review of travel-time prediction in transport and logistics', in *Proc. of the Eastern Asia Society for Transportation Studies*, volume 5, pp. 1433 – 1448, Hamburg, (2005).
- [14] G. Malnati, C. Barberis, and C.M. Cuva, 'Gossip: Estimating actual travelling time using vehicle to vehicle communication', in *Proc. of the 4-th Int. Workshop on Intelligent Transportation*, Hamburg, (2007).
- [15] C.E. McKnight, H.S. Levinson, C. Kamga, and R.E. Paaswell, 'Impact of traffic congestion on bus travel time in northern new jersey', *Transportation Research Record Journal*, **1884**, 27–35, (2004).
- [16] J.S. Racine, 'Consistent significance testing for nonparametric regression', *Journal of Business and Economic Statistics*, **15**, 369379, (1997).
- [17] B.L. Smith, B.M. Williams, and R.K. Oswald, 'Comparison of parametric and nonparametric models for traffic flow forecasting', *Transportation Research Part C*, **10**, 303–321, (2002).
- [18] S.S. Stankovic, M.S. Stankovic, and D.M. Stipanovic, 'Decentralized parameter estimation by consensus based stochastic approximation', *IEEE Trans. Automatic Control*, **56**, (2009).