Resampling based Modelling of Individual Routing Preferences in Distributed Traffic Network

Jelena Fiosina and Maksims Fiosins

Institute of Informatics
Clausthal University of Technology
Julius-Albert Str. 4, D-38678, Clausthal-Zellerfeld, Germany
{Jelena.Fiosina,Maksims.Fiosins}@gmail.com

ABSTRACT

Modern Internet technologies can considerably enhance contemporary intelligent transportation system (ITS). Such systems need to process increasing volumes of data available in clouds, and so new algorithms and techniques for statistical data analysis are required. A very important problem for ITS is optimal traffic routing, which requires modelling of individual routing preferences. This leads to the selection of the shortest itinerary, which requires route comparison on the basis of historical data and dynamic observations.s. This leads to the selection of the shortest itinerary, which requires route comparison on the basis of historical data and dynamic observations. In the paper, we propose a generic cloud-based system architecture, based on the collaboration of individual and cloud agents and resampling-based pairwise route comparison in a stochastic graph. The weights of the edges are considered to be independent random variables with unknown distributions. Only historical samples of the weights are available, and some edges may have common samples. We estimate the probability that the weight of the first route is greater than that of the second one. The analytical expressions for the expectations and variances allow theoretical evaluation of the method. This problem is very important for a vehicle decision-making procedure to choose route from the available alternatives. We apply a four-step decision-making process, instantiated for route recommendations and Markov chain based route ranking method for selection of the final decision. The experimental results demonstrate that the resampling estimates are more precise than parametric plug-in ones in the case of extreme small or extreme large sample sizes. This approach can be successfully used in route recommender systems.

Keywords: Bootstrap, jackknife and other resampling methods, Ranking and selection, Network models, stochastic Random graphs, Renewal theory, Markov chains, Agent technology, Distributed algorithms, Traffic problems.

2000 Mathematics Subject Classification: 62F40, 62F07, 90B15, 05C80 ,60K05, 60J10, 68T42, 68W15, 90B20.

1 Introduction

Modern Internet and communication opportunities open new perspectives on the development of intelligent transport systems (ITS). Technologies such as cloud and grid computing, the

Internet of Things concept and ambient intelligence methods allow the development of new applications, to hide the complexity of data and algorithms in the network. This allows traffic participants to run simple applications on their mobile devices, which provide clear recommendations on how they should act in the current situation. These simple applications are based on the aggregation and processing of large amounts of data, which are collected from various traffic participants and sensors. These data are physically distributed and available in virtual clouds. This creates a need for innovative data analysis, processing, and mining techniques, which run in clouds and prepare necessary information for end-user applications.

In this study, we deal with route recommender system, which is essential applications in ITS. This system includes optimization of the booked itinerary with respect to user preferences, time, fuel consumption, cost, and air pollution to provide better (i.e., quicker, more comfortable, cheaper, and greener) mobility. The recommendations are made on the basis of static information about the network (traffic lights, public transport schedules, etc.) combined with dynamic information about the current situation and historically stored data about traveling under equivalent conditions. If necessarily, the recommendations of other travelers. can be included. Booking the shortest itinerary is a key aspect in many traffic scenarios with different participants: a dynamic multi-modal journey, a simple private drive through a transport network, or smart city logistical operations. We consider an example of driving through a transport network segment considering the time consumption as the optimization criterion in itinerary comparisons and shortest route selection. In this case, the route recommendation is based on the estimates of the travel time along the route.

For this purpose, a simulated transport network is created, the travel times for alternative routes are estimated, and the best route is selected. Different methods of travel-time forecasting are used, such as regression models, and neural networks. Most of these are sensitive to outliers or incorrect model selection (e.g. wrong distribution). In these situations, the methods of computational statistics can be effective.

Computational statistics includes a set of methods for non-parametric statistical estimation. The main idea is to use data in different combinations to replace complex statistical inferences by computations. The resampling approach supposes that the available data are used in different combinations to obtain model-free estimators that are robust to outliers. The quality of the estimators obtained is also important.

In this study, we propose a generic cloud-based system architecture, based on the collaboration of individual and cloud agents and resampling-based pairwise route comparison in a stochastic graph. We apply a four-step decision-making process, instantiated for route recommendations and Markov chain based route ranking method for selection of the final decision. We derive the properties of the proposed resampling estimators and compare these with parametric plug-in estimators.

The remainder of this study is organized as follows: Section 2 presents state of the art. Section 3 formulates the problem, Section 4 describes resampling algorithm for cooperative route selection procedure. Section 5 describes the properties of resampling algorithm, Section 6 presents a Markov chain based ranking algorithm for routes selection. Section 6 illustrates the proposed approach with a numerical example, and Section 6 concludes the paper.

2 State of the Art

2.1 Big Transportation Data

Ubiquitous traffic sensors and the Internet of Things (Xiao and Wang, 2011) create world-wide network of interconnected objects uniquely addressable that ensure an exchange and sharing of information and an ability to interact with each other and cooperate with their neighbours to reach common goals in intelligent transportation system (ITS). One can speak about Big Data, which include massive data sets with sizes beyond the ability of commonly used tools to capture, curate, manage, and process the data within a tolerable elapsed time. Big Data (*Gartner Reveals Top Predictions for IT Organisations and Users for 2013 and Beyond*, 2013) are defined as high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight, decision making, and process optimization.

Computational resources, in their turn, are shifting toward parallel and distributed architectures, with multicore and cloud computing platforms providing access to hundreds or thousands of processors, which present new capabilities for storage and manipulation of data. However, from an inferential point of view, it is not yet clear how statistical methodology will transport to a world involving massive data on parallel and distributed computing platforms (Kleiner, Talwalkar, Sarkar and Jordan, 2012).

Providing business opportunities, Big Data means also major research challenges. One of the key challenges is connected with techniques for analyzing Big Data, which include distributed DA (DDA) supported by cloud computing power. By nature, Big Data is decentralized and should be properly analysed in decentralized fashion without transmission of big information volumes. DDA techniques should be modified for large-scale, streaming Big Data.

Centralization of DA methods supposes data aggregation in space and time, which is usually not feasible. Using the centralized approach the system cannot adapt quickly to situations in real time, and it is very difficult or simply impossible to transmit a large amount of information over the network and to store, manage and process massive data sets in one location (Fiosins, Fiosina, Müller and Görmer, 2011b). Moreover, some nodes of the distributed system prefer relay mostly of their own experience making forecasting process more autonomous.

2.2 Computational Statistics for Distributed Systems

Physical and logical distribution of constantly updated data storages in complex stochastic systems can be well represented by multi-agent system (MAS) architecture. The key challenges in MASs is the capability of agents to analyse distributed data sources and to provide sufficient information for optimal decisions (Freitas, 2002), (Symeonidis and Mitkas, 2005). Fully embedding information processing, described in terms of ubiquitous intelligence (Cao, Luo and Zhang, 2009), can be reached by coupling of MAS with DA. DA improve agent intelligence (da Silva, Giannella, Bhargava, Kargupta and Klusch, 2005), involving pro-active and autonomous agents that perceive their environment, dynamically reason out actions, and interact with each other. The knowledge of agents is a result of the outcome of empirical DA in addition to the pre-existing domain knowledge (Klusch, Lodi and Moro, 2003). This collective 'intelligence' of MAS must be developed by distributed domain knowledge and collaborative analysis of the distributed data observed by different agents (da Silva et al., 2005), (Zhang, Zhang and Cao, 2005).

Cloud computing can offer a very powerful, reliable, predictable and scalable computing infrastructure for the execution of MASs by implementing complex, agent-based applications for modeling and simulation. Agents can be used as basic components for implementing intelligence in clouds, making them more adaptive, flexible, and autonomic in resource management, service provisioning and large-scale application executions (Talia, 2011). This approach allows virtual centralisation, storing and management of such data using cloud computing technologies and logical decentralisation of data sources by MAS and their decentralised analysis and processing, including parallel cooperative computation. Data sources remain distributed, but connection between them becomes easier, the communication is broadly available, but a bottleneck is computation. For DDA methods using cloud computing the quality of the information is of great importance and they should know which information and where is available. However the problem of the information cost (speed of its extraction, quality, reliability, etc.) is prior the information availability. In this study, we will use MAS to represent DDA, taking special attention to development of algorithms which coordinate the distributed parts of system and synchronise the separate models and phases, represented by agents. In complex stochastic systems as ITS, many different factors should be estimated and traffic flows should be properly modelled and forecasted. MAS-based representation of ITS helps to overcome the limitations of centralised DA, which allows vehicles making decisions autonomously (Bazzan and Klügl, 2013). Therefore, new DDA methods should be developed and integrated into the existent ITS at different stages of its functioning, which are capable properly aggregate, filter and process Big Data.

A progress in Internet and communication technologies (ICT) facilitates data collection and provides the necessary resources for the operation of the computationally intensive methods of computational statistics (CST). This set of methods is the interface between statistics and computer science. CST is aiming at the design of algorithms for implementing statistical methods on computers, including the ones unthinkable before the computer age (e.g.; bootstrap, simulation) as well as to cope with analytically intractable problems. CST supposes an application of iterative calculations instead of complex analytical models and statistical procedures by using available data in different combinations. The resulting solution is approximate; however in many practical situations (too big or too small samples, complex and hierarchical structure of analyzed system, dependency in data) this may give more robust and precise results as classical methods or even provide a solution in the situations where classical methods fail. The term CST refers to computationally intensive statistical methods including resampling, bootstrap, cross-validation, Markov Chain Monte Carlo, non-parametric regression, kernel density estimation, generalized additive models, etc. CST approach can be successfully used in various DA techniques: forecasting models (Afanasyeva and Andronov, 2006), (Afanasyeva, 2005b), (Wu, 1986), clustering (Hinneburg and Gabriel, 2007) change-point analysis (Fiosina and Fiosins, 2011).

CST methods would seem ideally suited to straightforwardly leveraging parallel and distributed computing architectures: one might imagine using different processors or compute nodes to process different resamples independently in parallel. 'The Bag of Little Bootstraps' procedure (Kleiner et al., 2012) incorporates features of both the bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators.

A 'Divide and conquer' approach for distributed data analysis (Chen, 2013) assumes to divide the data of size n into K subsets of size $O(n/K)$. For each subset of data, they perform a DA and the results of DA of each of the K subsets are then combined to obtain an overall result. This approach is implemented for the situation that n is extraordinarily large, too large to perform the aforementioned DA using a single computer or available computing resources (Chen, 2013).

In the massive data setting, computation of even a single point estimate on the full dataset can be quite computationally demanding, and so repeated computation of an estimator on comparably sized resamples can be prohibitively costly. However, the large size of bootstrap resamples in the massive data setting renders this approach problematic, as the cost of transferring data to independent processors or compute nodes can be overly high, as is the cost of operating on even a single resample using an independent set of computing resources (Kleiner et al., 2012). Working with Big Data CST methods give a possibility to work with a parts of data in different combinations [samples from Big Data] in various applications (image recognition, particle filtering and artificial population), which make possible to process it with a tolerable elapsed time.

For streaming data, CST methods provide data pre-processing by selection resamples of data and obtaining representative samples is the only reasonable way to analyze the data (Rajaraman and Ullman, 2011). Data filtering method based on targeted sequential resampling and model mixtures of distributions using Markov chain Monte Carlo method was introduced in (Manolopoulou, Chan and West, 2010). In this study, we are planning to apply the described parallel calculation with CST methods in cloud-based infrastructure.

The traditional plug-in approach to the estimation of the probability of interest is a parametric one. It supposes: (1) to choose distribution type; (2) to calculate a point estimator of the parameters of the chosen distribution. In the case of small samples, it is difficult to choose the distribution law correctly; hence the estimators obtained are usually inaccurate.

Hence, it is preferable to use the non-parametric resampling procedure (Gentle, 2002), which is a variant of the bootstrap method (Davison and Hinkley, 1997), (Efron and Tibshirani, 1993). The implementation of this approach to various problems was considered in the studies reported in (Afanasyeva, 2005a), (Andronov, Fioshina and Fioshin, 2009), (Fioshin, 2000). We employ the usual simulation technique with one distinguishing feature: we don't make any parameter estimations, but extract elements in the simulation process randomly from the samples of random variables. We produce a series of independent experiments and accept the average over all realizations as the resampling estimator of the parameter of interest.

2.3 Probabilistic Preference Modelling

2.3.1 Travel Time Forecasting: Predictive models

Travel times play an important role in transportation and logistics. From travellers' viewpoints, the knowledge about travelling time helps to reduce delays and improves reliability through better selection of routes. In logistics, accurate travelling time estimation could help to reduce transport delivery costs and to increase the service quality of commercial delivery by bringing goods within the required time window. For traffic managers, travelling time is an important index for traffic system operation efficiency (Lin, Zito and Taylor, 2005).

There are several studies in which a centralised approach is used to predict travel times. The approach was used in various ITS, such as in-vehicle route guidance and advanced traffic management systems. A good overview is given in (Lin et al., 2005). To make the approach effective, agents should cooperate with each other to achieve their common goal via so-called gossiping scenarios. The estimation of the actual travelling time using vehicle-to-vehicle communication without MAS architecture was described in (Malnati, Barberis and Cuva, 2007).

A combination of centralized and decentralized agent-based approaches to the traffic control was presented in (Görmer, Ehmke, Fiosins, Schmidt, Schumacher and Tchouankem, 2011). In this approach, the agents maintain and share the 'local weights' for each link and turn, exchanging this information with a centralized traffic information centre. The decentralised MAS approach for urban traffic network was considered also in (Claes and Holvoet, 2011), 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 forecasting model.

A promising approach to agent-based parameter estimation for partially heterogeneous data in sensor networks was suggested in (Guestrin, Bodik, Thibaux, Paskin and Madden, 2004). Another decentralised approach for homogeneous data was suggested in (Stankovic, Stankovic and Stipanovic, 2009) to estimate the parameters of a wireless network by using a parametric linear model and stochastic approximations.

A problem of decentralised travel time forecasting was considered in (Fiosina, 2012), (Fiosina and Fiosins, 2012), (Fiosina and Fiosins, 2013a). An MAS-based architecture with autonomous agents was implemented for this purpose. A decentralised linear (Fiosina, 2012), (Fiosina and Fiosins, 2013a) and kernel density (KD) based (Fiosina and Fiosins, 2012), (Fiosina and Fiosins, 2013a) multivariate regression models were developed to forecast the travelling time. The iterative least square estimation method was used for regression parameter estimation, which is suitable for streaming data processing. The resampling-based consensus method was suggested for coordinated adjustment of estimates between neighbouring agents. The efficiency of the suggested approach using simulation with data from the southern part of Hanover was illustrated. The experiments showed the efficiency of the proposed approach. The prediction technique in tutorial style was described in terms of distributed network intelligence in (Fiosina and Fiosins, 2013a). The comparison of parametric and non-parametric approaches for traffic-flow forecasting made in (Smith, Williams and Oswaldl, 2002), demonstrates the efficiency of the non-parametric KD regression (Fiosins, Fiosina, Müller and Görmer, 2011a), (Fiosins et al., 2011b).

2.3.2 Traffic routing problem

A traffic routing problem with decentralized decision making of vehicle agents in urban traffic system was investigated, where the planning process for a vehicle agent is separated into two stages: strategic planning for selection of the optimal route and tactical planning for passing the current street in the optimal manner. A MAS architecture and the necessary computational statistics-based algorithms for comparing two routes in a stochastic graph (Fiosins et al., 2011a), (Fiosins et al., 2011b), and the shortest path search were developed (Fiosina and Fiosins, 2013b), which are carried out at strategic planning stage. The models were implemented to real data and integrated into a traffic domain application use case, where efficiency of the algorithms was evaluated. Distributed optimization approach for traffic flow routing was considered in (Fiosins, 2013). A combination of centralized and decentralized agent-based approaches to the traffic control was presented in (Görmer et al., 2011), where the agents maintain and share the 'local weights' for each link and turn, exchanging this information with a centralized traffic information centre.

3 Problem Formulation

3.1 System Architecture

We consider a cloud-based ITS architecture (Li, Chen and Wang, 2011). In terms of the Internet of Things, the real-world users are represented in the cloud system as virtual agents, which act in the cloud and virtual traffic network. The street network is presented by the simulated transport network, which consists of a digital map as well as the associated ad-hoc network models that allow estimation and forecasting of the important network characteristics for each problem (Fiosins et al., 2011b). The virtual agents store the real-time information, which is collected and constantly processed in the cloud. Moreover, the strategies for execution of the cloud application are constantly pre-calculated and checked in the virtual network (e.g., the shortest routes are pre-calculated). When a user runs the cloud application, the pre-calculated strategy is updated with the real-time data and is executed, with respect to the corresponding changes. Data flows and corresponding optimization methods in the cloud-based ITS architecture are presented in Fig. 1.

We consider an application that provides route recommendations to vehicle drivers. The essential process of this application is the comparison of pre-defined routes. It is based on historical samples of the route segments, which are collected from the virtual users. The candidate routes are compared in the virtual transport network in order to recommend the best route to a user.

The considered system consists of two levels: individual level and cloud level. The users of the system are constantly connected to the cloud and have their virtual representations - agents on the individual level. Each agent in the cloud system represents a physical traffic participant; it has a number of goals defined by the user and relative freedom of means to achieve this

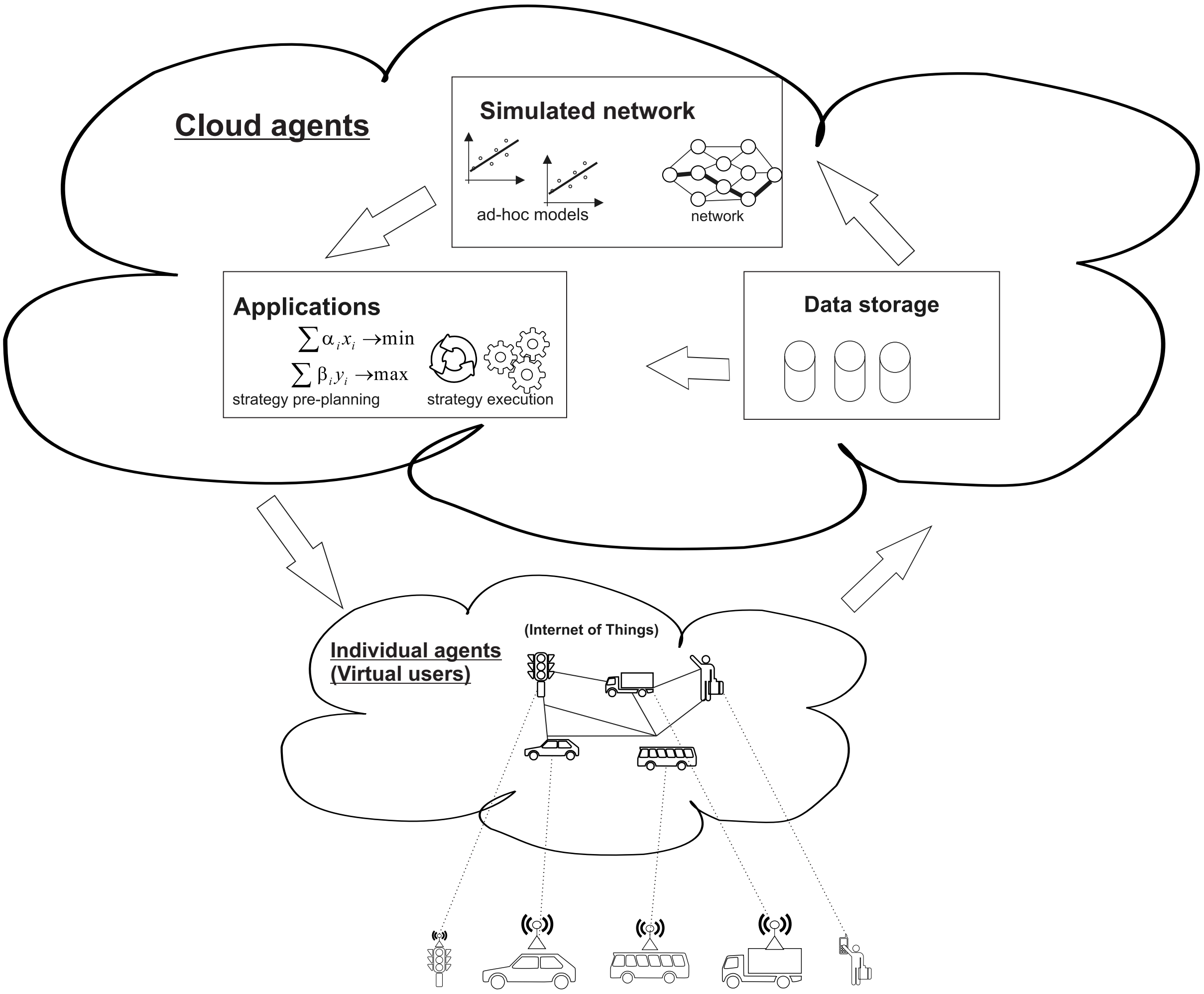

Figure 1: Data flows and corresponding optimization methods in ITS

goal. For example, if a goal of the agent is to plan (and correspondingly, to follow) an optimal route under certain conditions and constraints defined by the user, the agent can communicate or cooperate with other agents in the system. However important decisions (such as final route selection from a set of alternatives or cooperation in driving) makes a human.

There is another type of agents - cloud agents, which goal is to provide services and support to the individual level agents in their problem-solving. Cloud level agents are virtual central agents, which have a global picture of the system and can provide, for example, route recommendations to the individual agents.

All the agents of the system, both on the individual level and on the cloud level have a similar four-step decision-making loop demonstrated in Fig. 2 (left).

Let us consider the mentioned steps in detail.

- On the information collection and storage step, agents collect information from the environment and put it into virtual storages, depending on a type of the agent. For example, individual agents act as data generators, because they have physical sensors and are associated with physical data (location, speed etc.). The users decide, which part of the information should be stored in the cloud. Individual agents act as data sources for the cloud agents, which perform mining of relevant data and organize virtual repositories of

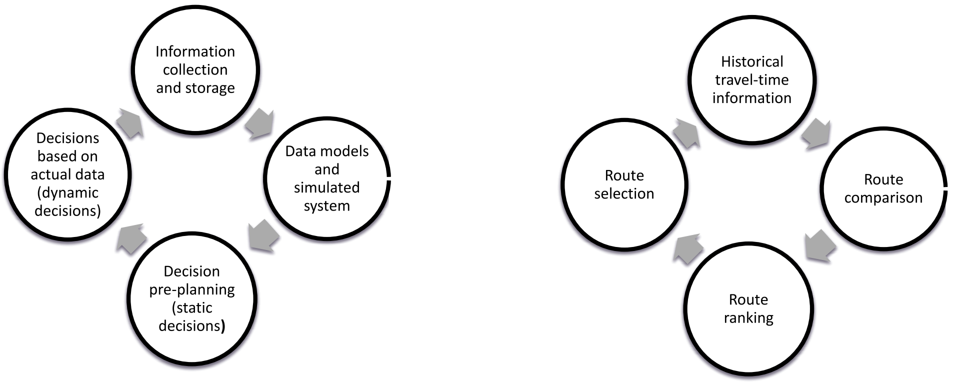
Information collection and storage
Data models and simulated system
Decision pre-planning (static decisions)
Decisions based on actual data (dynamic decisions)
Historical travel-time information
Route comparison
Route ranking
Route selection

Figure 2: Generic four-stage decision loop of the agents (left) and its example for the routing problem (right)

the relevant data.

- On the data models step, agents construct relevant data models to get and forecast important system characteristics (e.g. travel time). If this information cannot be directly calculated or forecasted, a simulation-based approach can be used for system modelling. Individual agents rely on their own experience and deal with a limited amount of data, which correspond to the local view to the system. However they can apply relatively complex data models or simulations, because the cloud provides relevant computational resources. Cloud agents have global view to the system and so should deal with big amounts of information (Big Data). They also apply complex data models or simulations to get the necessary system characteristics.

- On the static decision step, agents pre-calculate decisions for different possible situations (e.g. travel time under different conditions or for different sources-destinations). This creates a decision library, which can be used for a fast decision-making. These decisions can be checked on the simulation model as well.

- On the dynamic decision step, agents make decisions based on actual system conditions.

The described above decision-making procedure for the route selection problem is illustrated in Fig. 2 (right).
Let us consider this interpretation.

- Initial data is represented by travel times over the street network. This information is collected by vehicles and associated in the cloud with the corresponding individual agents. Cloud agent mines and collects this data to create its own virtual repository of the travel-time data.

- The graph and the resampling-based route comparison algorithm (Section 4.1) are used as a data model. Resampling allows to solve the problem of Big Data, especially for cloud agents.

- The Markov-chain based ranking algorithm (Section 4.2) is used to rank the routes based on the comparison results. The outcome of the algorithm is a probability distribution over the routes, which is used for route selection
- At the actual route selection stage the cloud agents provide to the individual agents their recommended route distribution (or a part of it). The individual agents make a final route decision based on their own probability distribution of routes and the distribution received from cloud agents.

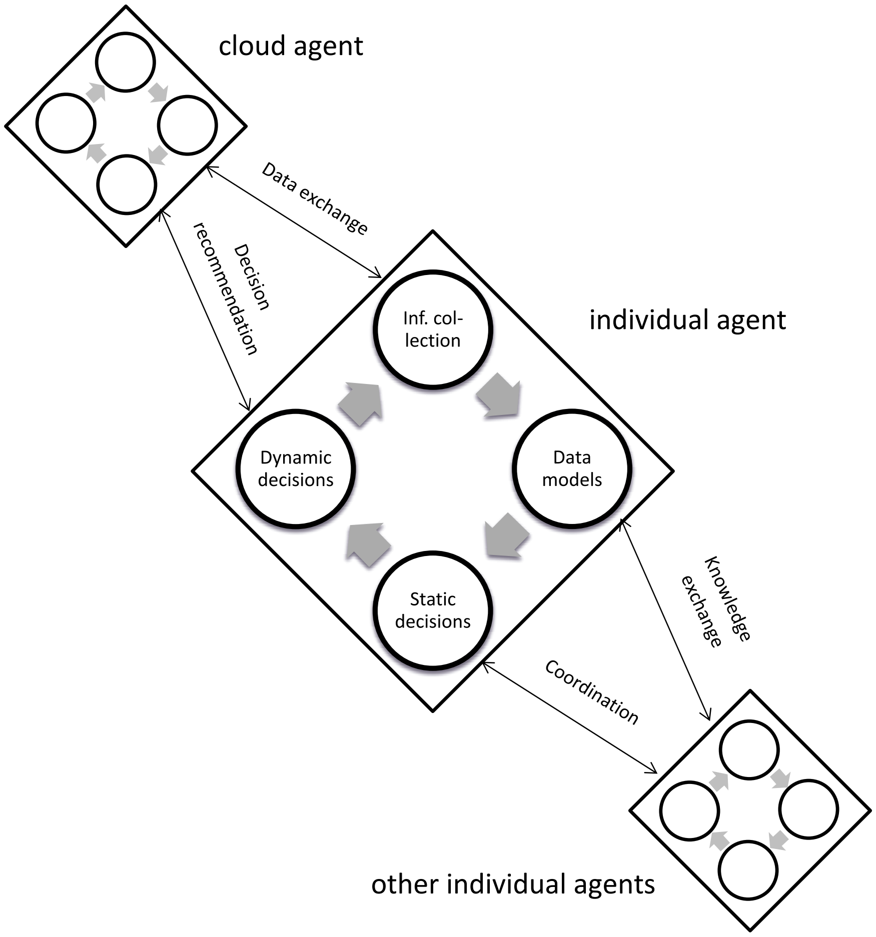

Figure 3: Decision loop for the route selection and cooperation between agents

In the system, agents can cooperate at any stage of their decision-making process. However the most important cooperations are following (Fig. 3):

- Individual and cloud agents cooperate mostly on the data collection and dynamic decision steps. On the data collection step, individual agents act as raw data sources for the cloud agent. Data model construction and static decision-making process are different for individual and the cloud agents as they act with different system views. They cooperate again on the dynamic decision-making step only, when the cloud agent provides a decision recommendation and the individual agent uses it in its individual decision-making process.

- Cooperation between individual agents is mostly on data models and static decision step. On these two steps, agents can adjust parameters of their local data models for more precise forecasting or their plans for the common planning (coordination).

3.2 Formal definitions

We consider a directed graph $G = (V,E)$ with n edges, $|E| = n$, where each edge $e_i \in E$ has an associated weight X_i (e.g. travel time). We assume that the weights $\{X_1,X_2,\ldots,X_n\}$ are independent random variables (r.v.).

A route in the graph is a sequence of edges such that the next edge in the sequence starts from the node, where the previous edge ends. We consider a set of routes $R = \{r^1,r^2,\ldots\}$ (they possibly have the same source and destination, but this is not essential for our procedure). A route r^b, $b = 1,2,\ldots$ is defined by a sequence k^b of edge indices in the initial graph $k^b = (k^b_1,k^b_2,\ldots,k^b_{n_b})$, which consists of n_b edges. Hence, a route r^b is the sequence of edges: $r^b = \{e_{k^b_1},e_{k^b_2},\ldots,e_{k^b_{n_b}}\}$. The route weight S^b is the sum of the corresponding edge weights, so $S^b = \sum_{i\in k^b} X_i$.

The distributions of the edge weights are unknown. This means that the weights (e.g. travel times) are collected in samples, but the distribution low is neither assumed nor estimated. Such information is local for each agent in the system. Let us define $H^a_i = \{H_{i,1},H_{i,2},\ldots,H_{i,m^a_i}\}$ the sample, which contains observations collected by the agent a about the weights of edge i, $i = 1,2,\ldots,c$, $c \leq n$. Note that m^a_i can be very big as well as very small. An (unknown) true cumulative distribution function (cdf) of the sample H_i elements is denoted by $F_i(x)$, $i = 1,2,\ldots,c$, and the elements of samples H^a_i for all a have the same distribution $F_i(x)$, which means that all agents observe the same system.

Note that we consider a situation, when a sample may correspond to several edges. This can a case in the situation, when two similar edges are observed and these observations are not separated and collected in one sample; as well, this can be in the situation when no observations about an edge are available and a sample of observations about another edge is used instead. So each sample may correspond to one or several edges and a number of samples c can be less that the number of edges n.

On the second stage of the four-step route selection process, each agent performs pairwise comparisons of routes based on available samples H^a_i. This means that the probabilities $p^{*a}_{b,b'}$ that the route b has bigger weight than the route b' are estimated:

$$p^{*a}_{b,b'} = P^*\{S^b > S^{b'}\}.$$

Note that the estimates $p^{*a}_{b,b'}$ are consistent estimators of true probabilities $p_{b,b'} = P\{S^b > S^{b'}\}$ ($\lim_{m^a_i\to\infty} p^{*a}_{b,b'} = p_{b,b'}$). For calculation of $p^{*a}_{b,b'}$ we use a resampling procedure described in Section 4.1.

The next problem is routes ranking. We use Markov chain based ranking, described in the Section 4.2 and calculate the probability distribution π^a over the routes R.

Finally, the agent decides about a route. It has a distribution π^a and receives a recommendation $\pi^{a'}$ from the cloud agent a'. A decision idea is to create a mix of distributions π^a and $\pi^{a'}$. The agent uses a constant $0 \leq \alpha^a \geq 1$. The following two-step procedure is used:

- A distribution π^a is selected with a probability α and a distribution $\pi^{a'}$ with a probability $1-\alpha$;

- A route is selected according to the selected distribution.

4 Cooperative Route Selection

4.1 Resampling Procedure for Route Preference Estimation

Let us consider a procedure for a pairwise comparison of two non-overlapping routes. In general formulation of Section 3.2 we compare the routes r^b and $r^{b'}$. For simplicity in this Section we suppose that $b=1$ and $b'=2$, so we compare the routes with indices 1 and 2. So our purpose is to calculate the probability that the weight of route 1 is greater than that of route 2:

$$\Theta = P\{S^1 > S^2\}. \tag{4.1}$$

Two cases are considered: (1) each edge has different samples, so only one element is extracted from the sample H_i; and (2) edges may correspond to common samples, including the common samples for two routes.

We propose an N-step resampling procedure. At each step, we randomly without replacement choose $\eta_i^1+\eta_i^2$ elements from each sample H_i: η_i^1 elements for route 1, and η_i^2 elements for route 2: $\eta_i = (\eta_i^1, \eta_i^2)$.

Let $J_i^b(l)$, $|J_i^b(l)| = \eta_i^b$ be a set of element indices extracted from the sample H_i, for a route b, $b=1,2$, during resampling step l, $i=1,\ldots,c$.

Let

$$\mathbf{X}^{*l} = \bigcup_{i=1}^{c}\{H_{i,j} : j \in J_i^1(l)\} \cup \bigcup_{i=1}^{c}\{-H_{i,j} : j \in J_i^2(l)\} \tag{4.2}$$

be the l-th resample of the edge weights for both routes, with the weights of route 2 assumed to be negative.

Let $\Psi(\mathbf{x})$ be an indicator function, where $\mathbf{x} = (x_1, x_2, \ldots)$ is a vector of real numbers: $\Psi(\mathbf{x})$ is unity if $\sum\limits_i x_i > 0$; otherwise, it is zero.

The average of $\Psi(\mathbf{X}^{*l})$ over all N steps is accepted as the resampling estimator of the probability of interest:

$$\Theta^* = \frac{1}{N}\sum_{l=1}^{N}\Psi(\mathbf{X}^{*l}). \tag{4.3}$$

The resampling-based route comparison procedure is presented in Algorithm 1.

The function $extract(X,n)$ randomly chooses n elements without replacement from the set X. The function $subsample(X,a,n)$ returns n elements from X, starting from position a. These two cases differ with the parameters of the $extract$ procedure.

4.2 Markov Chain Based Ranking of Routes

Now we consider an algorithm for ranking of routes based on Markov Chain (Negahban, Oh and Shah, 2012). For this purpose, agent a constructs a Markov chain $M^a = (R, E^a, P^a)$, where

Algorithm 1 Function RESAMPLING COMPARE

```
function RESAMPLING COMPARE($H_i, \eta_i, i = 1, \dots, c$, $N$)
    for all $l \in 1, \dots, N$ do
        for all $i \in 1 \dots c$ do
            $X_i^{*l} \leftarrow\ extract(H_i, \eta_i^1 + \eta_i^2)$
            $X1_i^{*l} \leftarrow\ subsample(X_i^{*l}, 1, \eta_i^1)$; $X2_i^{*l} \leftarrow\ subsample(X_i^{*l}, \eta_i^1 + 1, \eta_i^2)$
        end for
        $\mathbf{X}^{*\mathbf{l}} = \bigcup X1_i^{*l} \bigcup -X2_i^{*l}$; $\Theta_l \leftarrow \Psi(\mathbf{X}^{*\mathbf{l}})$
    end for
    $\Theta^* \leftarrow \frac{1}{N} \sum_{l=1}^{N} \Theta_l$
    return $\Theta^*$
end function
```

the states are considered routes $R = \{r^1, r^2, \dots\}$, transitions $E^a \subset R \times R$ are a set of route pairs being compared by the agent a. Let $P^a = \{p^a_{b,b'} | (r^b, r^{b'}) \in E^a\}$ be outcomes of comparisons, $d_{\max}$ be maximum out-degree of a node. The transition probabilities P of this Markov chain are defined as

$$p_{ij} = \begin{cases} 1/d_{\max} a_{i,j} & \text{if } i \neq j, \\ 1 - 1/d_{\max} \sum a_{i,k} & \text{if } i = j. \end{cases} \tag{4.4}$$

There are some intuitive arguments, why the Markov chain can be a good model for analysis of route comparisons. We suppose that if some route was selected by the decision-maker, at the next time moment a route preferred by the comparisons will be selected. So we represent the semantics of comparisons as transitions from one route to another.

Assume that the considered Markov chain is irreducible, which means that the graph (R,E) is connected. Let us require as well that at least one state is aperiodic (e.g. has a loop edge). The first condition can be achieved by proper selection of compared routes (the set E). The second condition is achieved by construction of the transition probabilities P (4.4). In this case the considered chain is ergodic and have an unique stationary distribution. Stationary distribution represents a fraction of a time, which will be spent in each state. In our interpretation, the routes, which has bigger values of stationary distribution, will be more preferred.

Let $p_t(i) = P\{X_t = i\}$ denote the distribution of the Markov chain at time t, where X_t is a state of the chain, $p_t = \{p_t(i)\}$. Then it is well-known that

$$p_{t+1} = p_t P.$$

Also there exists an unique stationary distribution $\pi = \lim_{t \to \infty} p_t$, which does not depend on the initial distribution p_0. A possible way to calculate it is to solve a system of equations

$$(P^T - I)\pi = 0$$

(for non-trivial π and requiring $\sum \pi = 1$).

The stationary distribution π provides ranking for the routes R. Moreover, the distribution itself will be used as a desired distribution of vehicles among the routes in the system.

5 Properties of the Resampling Algorithm

The estimator Θ^* is obviously unbiased: $E(\Theta^*) = \Theta$, so we are interested in its variance. Consider the elements extracted at two different steps $l \neq l'$. Moreover, we denote:

$$\mu = E\ \Psi(\mathbf{X}^{*l}),\ \mu_2 = E\ \Psi(\mathbf{X}^{*l})^2,\ \mu_{11} = E\ \Psi(\mathbf{X}^{*l}) \cdot \Psi(\mathbf{X}^{*l'}),\ l \neq l'. \tag{5.1}$$

Then, the variance of the estimator 4.3 is

$$V(\Theta^*) = E(\Theta^{*2}) - \mu^2 = \left\{ \frac{1}{N}\mu_2 + \frac{N-1}{N}\mu_{11} \right\} - \mu^2, \tag{5.2}$$

for the estimation of which we need the mixed moment μ_{11} depending on the resampling procedure.

5.1 Different Samples for Each Edge

In this case, $J_i^b(l)$ consists of one element, denoted as $j_i^b(l)$. This is the index of an element extracted from the sample H_i at step l for route b.

Let $M_i = \{1, 2, \ldots, m_i\}$, $U^b : \{i : \eta_i^b \neq \emptyset\}$, $M^b = \prod_{i \in U^b} M_i$ and

$$\begin{aligned} &\mathbf{j}^b(l) = \{j_i^b(l) : i \in U^b\}, \\ &\mathbf{j}(l) = (\mathbf{j}^1(l), \mathbf{j}^2(l)), \end{aligned} \tag{5.3}$$

where $\mathbf{j}^b(l) \in M^b$ and $b = 1, 2$.

We use a modification of the ω-pair notation (Fioshin, 2000). Let $\omega^b \subset U^b$, $\omega = (\omega^1, \omega^2)$. We assume that two vectors $\mathbf{j}(l)$ and $\mathbf{j}(l')$ produce an ω-pair, if $j_i^b(l) = j_i^b(l')$ for $i \in \omega^b$ and $j_i^b(l) \neq j_i^b(l')$ for $i \notin \omega^b$. In other words, the components of the vectors $\mathbf{j}(l)$ and $\mathbf{j}(l')$ produce the ω-pair if they have the same elements from the samples, whose indices are contained by ω.

Let $A(\omega)$ be an event 'resamples $\mathbf{j}(l)$ and $\mathbf{j}(l')$ for the different steps $l \neq l'$ produce the ω-pair', let $P\{\omega\}$ be the probability of this event, and let $\mu_{11}(\omega)$ be the corresponding mixed moment. The probability of producing the ω-pair is

$$P\{\omega\} = \frac{1}{|M^1||M^2|} \prod_{i \in \bigcup_b \{U^b \setminus \omega^b\}} (m_i - 1). \tag{5.4}$$

The mixed moment μ_{11} can be calculated with the formula

$$\mu_{11} = \sum_{\omega \subset U^1 \times U^2} P(\omega)\mu_{11}(\omega). \tag{5.5}$$

Next, we intend to calculate $\mu_{11}(\omega)$, $\omega \subset U^1 \times U^2$. Let

$$\begin{aligned} &S_l^{dif}(\omega) = \textstyle\sum_{i \in U^1 \setminus \omega^1} H_{i, j_i^1(l)} - \sum_{i \in U^2 \setminus \omega^2} H_{i, j_i^2(l)}, \\ &S_{ll'}^{com}(\omega) = \textstyle\sum_{i \in \omega^1} H_{i, j_i^1(l)} - \sum_{i \in \omega^2} H_{i, j_i^2(l)}. \end{aligned} \tag{5.6}$$

We note also, that the mentioned sums can be expanded in another way as follows

$$\begin{aligned} \sum_{i \in U^1} H_{i, j_i^1(l)} &= \sum_{i \in U^1 \setminus \omega^1} H_{i, j_i^1(l)} + \sum_{i \in \omega^1} H_{i, j_i^1(l)}, \\ \sum_{i \in U^2} H_{i, j_i^2(l)} &= \sum_{i \in U^2 \setminus \omega^2} H_{i, j_i^2(l)} + \sum_{i \in \omega^2} H_{i, j_i^2(l)}. \end{aligned} \tag{5.7}$$

Then, $\mu_{11}(\omega)$ can be calculated as

$$\begin{aligned}
&\mu_{11}(\omega) = E(\Psi(\mathbf{X}^{*l}) \cdot \Psi(\mathbf{X}^{*l'})|\omega) = P\left\{\Psi(\mathbf{X}^{*l}) = 1, \Psi(\mathbf{X}^{*l'}) = 1|\omega\right\} = \\
&= P\left\{\sum_{i \in U^1} H_{i,j_i^1(l)} - \sum_{i \in U^2} H_{i,j_i^2(l)} > 0, \sum_{i \in U^1} H_{i,j_i^1(l')} - \sum_{i \in U^2} H_{i,j_i^2(l')} > 0|\omega\right\} = \\
&= P\left\{S_l^{dif}(\omega) + S_{ll'}^{com}(\omega) > 0, S_{l'}^{dif}(\omega) + S_{ll'}^{com}(\omega) > 0\right\} = \\
&= \int_{-\infty}^{+\infty} P\left\{S_l^{dif}(\omega) > -x\right\} \cdot P\left\{S_{l'}^{dif}(\omega) > -x\right\} dF_{\omega(x)}^{c} = \int_{\infty}^{+\infty} \left(1 - F_\omega^d(-x)\right)^2 dF_{\omega(x)}^{c},
\end{aligned} \tag{5.8}$$

where $F_\omega^d(x)$ is cdf of $S_l^{dif}(\omega)$, $F_\omega^c(x)$ is cdf of $S_{ll'}^{com}(\omega)$ given ω-pair. Note, that for the fixed value of r.v. $S_{ll'}^{com}(\omega) = x$ the events $\left\{S_l^{dif}(\omega) > -x\right\}$ and $\left\{S_{l'}^{dif}(\omega) > -x\right\}$ are independent.

5.2 Common Samples for Edges

Here, we use the notation of α-pairs (Afanasyeva, 2005a), (Andronov et al., 2009), (Fioshin, 2000) instead of ω-pairs.

Let

$$\begin{aligned}
&J_i^b(l) = \{j_{i,1}^b(l), j_{i,2}^b(l), \ldots, j_{i,\eta_i^b}^b(l)\}, \\
&J^b(l) = \{J_i^b(l) : i \in U^b\}, \\
&\mathbf{J}(l) = \{J^1(l), J^2(l)\},
\end{aligned} \tag{5.9}$$

where $J_i^b(l) \subset M^b$, $b = 1,2$, $l = 1, \ldots, N$, $i = 1, 2, \ldots, c$.

Let $A_i^b(ll')$ be a set of indices of the common elements, extracted from the sample H_i for route b at steps l and l'. Let $A_i^{bp}(ll')$ be a set of indices of the common elements, extracted from the sample H_i for route b at step l and for route p and at step l'. Let $\overline{A}_i^{bp}(l)$ be a set of indices of the elements from route b at step l, which were in neither route b nor route p at step l', $b, p \in \{1,2\}$ and $b \neq p$:

$$\begin{aligned}
&A_i^b(ll') = J_i^b(l) \cap J_i^b(l') \\
&A_i^{bp}(ll') = J_i^b(l) \cap J_i^p(l') \\
&\overline{A}_i^{bp}(l) = J_i^b(l) \setminus (A_i^b(ll') \cup A_i^{bp}(ll')) \\
&\overline{A}_i^{pb}(l) = J_i^p(l) \setminus (A_i^p(ll') \cup A_i^{pb}(ll')).
\end{aligned} \tag{5.10}$$

Let $0 \leq \alpha_i^b \leq \eta_i^b$, $0 \leq \alpha_i^{bp} \leq \min(\eta_i^b, \eta_i^p)$, $b, p \in \{1,2\}$ and $b \neq p$. Let $\alpha_i = \{\alpha_i^1, \alpha_i^2, \alpha_i^{12}, \alpha_i^{21}\}$, $\alpha = \{\alpha_i\}, i = 1, 2, \ldots, c$. Next, we say that $\mathbf{J}(l)$ and $\mathbf{J}(l')$ produce an α-pair, if and only if:

$$\begin{aligned}
&\alpha_i^1 = |A_i^1(ll')|, \\
&\alpha_i^2 = |A_i^2(ll')|, \\
&\alpha_i^{12} = |A_i^{12}(ll')|, \\
&\alpha_i^{21} = |A_i^{21}(ll')|.
\end{aligned} \tag{5.11}$$

Let $A_{ll'}(\alpha)$ denote the event 'subsamples $\mathbf{J}(l)$ and $\mathbf{J}(l')$ produce an α-pair', and let $P_{ll'}\{\alpha\}$ be the probability of this event: $P_{ll'}\{\alpha\} = P_{ll'}\{A_{ll'}(\alpha)\}$.

To calculate $\mu_{11}(\alpha)$ we replace ω-pairs with α-pairs. Therefore we need to calculate $P\{\alpha\}$ and $\mu_{11}(\alpha)$. The probability $P\{\alpha\}$ is

$$P\{\alpha\} = \prod_{i\in 1,2,\ldots,c} \frac{\binom{\eta_i^1}{\alpha_i^1}\binom{\eta_i^2}{\alpha_i^{21}}\binom{m_i-\eta_i^1-\eta_i^2}{\eta_i^1-\alpha_i^1-\alpha_i^{21}}}{\binom{m_i}{\eta_i^1}} \times$$
$$\times \binom{\eta_i^1-\alpha_i^1}{\alpha_i^{12}}\binom{\eta_i^2-\alpha_i^{21}}{\alpha_i^2} \frac{\binom{m_i-2\eta_i^1-\eta_i^2+\alpha_i^1+\alpha_i^{21}}{\eta_i^2-\alpha_i^{12}-\alpha_i^2}}{\binom{m_i-\eta_i^1}{\eta_i^2}},$$

where $\binom{n}{m}$ is a binomial coefficient.

To calculate $\mu_{11}(\alpha)$ we divide each sum into three subsums: $S_l^{dif}(\alpha)$ contains different elements for steps l and l'; $S_{ll'}^{com}(\alpha)$ - the common elements for the same route; $S_{ll'}^{com12}(\alpha)$ - the common elements for different routes.

Let

$$\begin{aligned} S_l^{dif}(\alpha) &= \sum_{i=1}^{c}\left\{\sum_{j\in \bar{A}_i^{12}(l)} H_{i,j} - \sum_{j\in \bar{A}_i^{21}(l)} H_{i,j}\right\}, \\ S_{ll'}^{com}(\alpha) &= \sum_{i=1}^{c}\left\{\sum_{j\in A_i^{1}(ll')} H_{i,j} - \sum_{j\in A_i^{2}(ll')} H_{i,j}\right\}, \\ S_{ll'}^{com12}(\alpha) &= \sum_{i=1}^{c}\left\{\sum_{j\in A_i^{12}(ll')} H_{i,j} - \sum_{j\in A_i^{21}(ll')} H_{i,j}\right\}. \end{aligned} \tag{5.12}$$

We note also, that the mentioned sums can be expanded in another way as follows

$$\begin{aligned} \sum_{i=1}^{c}\sum_{j\in J_i^1(l)} H_{i,j} &= \sum_{j\in \bar{A}_i^{12}(l)} H_{i,j} + \sum_{j\in A_i^{1}(ll')} H_{i,j} + \sum_{j\in A_i^{12}(ll')} H_{i,j}, \\ \sum_{i=1}^{c}\sum_{j\in J_i^2(l)} H_{i,j} &= \sum_{j\in \bar{A}_i^{21}(l)} H_{i,j} + \sum_{j\in A_i^{2}(ll')} H_{i,j} + \sum_{j\in A_i^{21}(ll')} H_{i,j}. \end{aligned} \tag{5.13}$$

As $S_{ll'}^{com}(\alpha) = S_{l'l}^{com}(\alpha)$ and $S_{ll'}^{com12}(\alpha) = -S_{l'l}^{com12}(\alpha)$, $\mu_{11}(\alpha)$ is:

$$\begin{aligned} &\mu_{11}(\alpha) = E\{\Psi(\mathbf{X}^{*l})\cdot\Psi(\mathbf{X}^{*l'})|\alpha\} = P\left\{\Psi(\mathbf{X}^{*l}) = 1, \Psi(\mathbf{X}^{*l'}) = 1|\alpha\right\} = \\ &= P\left\{\sum_{i=1}^{c}\left(\sum_{j\in J_i^1(l)} H_{i,j} - \sum_{j\in J_i^2(l)} H_{i,j}\right) > 0, \sum_{i=1}^{c}\left(\sum_{j\in J_i^1(l')} H_{i,j} - \sum_{j\in J_i^2(l')} H_{i,j}\right) > 0|\alpha\right\} = \\ &= P\left\{S_l^{dif}(\alpha) + S_{ll'}^{com}(\alpha) + S_{ll'}^{com12}(\alpha) > 0, S_{l'}^{dif}(\alpha) + S_{ll'}^{com}(\alpha) - S_{ll'}^{com12}(\alpha) > 0\right\} = \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (1 - F_\alpha^d(-x-y)) \times (1 - F_\alpha^d(-x+y)) dF_\alpha^c(x) dF_\alpha^{c12}(y), \end{aligned} \tag{5.14}$$

where $F_\alpha^d(x)$ is cdf of $S_l^{dif}(\alpha)$, $F_\alpha^c(x)$ is cdf of $S_{ll'}^{com}(\alpha)$, $F_\alpha^{c12}(x)$ is cdf of $S_{ll'}^{com12}(\alpha)$.

5.3 Special Case: Normal Distribution

In this section we illustrate the proposed approach for a case of normally distributed weights; samples H_i are normally distributed:

$$F_i(x) = \Phi\left(\frac{x-\beta_i}{\sigma_i}\right),$$

where $\Phi(x)$ is standard normal distribution function with mean 0 and variance 1.
The probability of interest can be represented by formula:

$$\begin{aligned}\Theta(\beta,\sigma) &= P\{S^1 > S^2\} = \\ &= 1-\Phi\left(\frac{0-\left(\sum\limits_{i\in U_1}\beta_i - \sum\limits_{i\in U_2}\beta_i\right)}{\sqrt{\sum\limits_{i\in U_1}\sigma_1^2 + \sum\limits_{i\in U_2}\sigma_i^2}}\right).\end{aligned} \tag{5.15}$$

The parametric plug-in approach supposes an estimation of the parameters β_i, σ_i to obtain $\tilde{\beta}_i$, $\tilde{\sigma}_i$ using the available sample populations. Then to calculate the estimator $\tilde{\Theta}(\tilde{\beta},\tilde{\sigma})$ we use formula (5.15) replacing β with $\tilde{\beta}$ and σ with $\tilde{\sigma}$.
Then the expectation of this estimator is:

$$\begin{aligned}E(\tilde{\Theta}) = &\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty}\cdots\int\limits_{-\infty}^{\infty}\tilde{\Theta}(x,\{y_1,\ldots,y_c\})\times \\ &\times f_{\tilde{\beta}_s}(x)f_{\tilde{\sigma}_1}(y_1)\ldots f_{\tilde{\sigma}_c}(y_c)dxdy_1\ldots dy_c,\end{aligned} \tag{5.16}$$

where

$$f_{\tilde{\beta}s}(x) = \Phi\left(\frac{\sum\limits_{i\in U_1}\beta_i - \sum\limits_{i\in U_2}\beta_i}{\sqrt{\sum\limits_{i\in U_1}\sigma_i^2/m_i + \sum\limits_{i\in U_2}\sigma_i^2/m_i}}\right)$$

is pdf of $\sum\limits_{i\in U_1}\tilde{\beta}_i - \sum\limits_{i\in U_2}\tilde{\beta}_i$, and $f_{\tilde{\sigma}_i}(x)$ is pdf of $m_i\tilde{\sigma}_i^2/\sigma_i^2$, which is χ^2 distributed with m_i degrees of freedom.
The expression for the second moment $E(\tilde{\Theta}^2)$ can be calculated by replacing in (5.16) $\tilde{\Theta}(\tilde{\beta},\tilde{\sigma})$ with $\tilde{\Theta}(\tilde{\beta},\tilde{\sigma})^2$. Then the variance $V(\tilde{\Theta})$ and mean squared error $MSE(\tilde{\Theta})$ of $\tilde{\Theta}$ are:

$$\begin{aligned}&V(\tilde{\Theta}) = E(\tilde{\Theta}^2) - E(\tilde{\Theta})^2, \\ &MSE(\tilde{\Theta}) = V(\tilde{\Theta}) + \left(\Theta - E(\tilde{\Theta})\right)^2.\end{aligned}$$

Now let us consider the resampling estimator. For the case, demonstrated in Section 5.1 distributions of the sums (5.6) are also normal:

$$\begin{aligned}F_\omega^d(x) &= \Phi\left(\frac{x-\sum\limits_{i\in U^1\setminus\omega^1}\beta_i - \sum\limits_{i\in U^2\setminus\omega^2}\beta_i}{\sum\limits_{i\in U^1\setminus\omega^1}\sigma_i + \sum\limits_{i\in U^2\setminus\omega^2}\sigma_i}\right), \\ F_\omega^c(x) &= \Phi\left(\frac{\sum\limits_{i\in\omega^1}\beta_i - \sum\limits_{i\in\omega^2}\beta_i}{\sum\limits_{i\in\omega^1}\sigma_i + \sum\limits_{i\in\omega^2}\sigma_i}\right).\end{aligned} \tag{5.17}$$

For the case from Section 5.2 the sums (5.12) distributions are normal:

$$\begin{aligned} F^d_\alpha(x) &= \Phi\left(\frac{\sum\limits_{i=1}^{c}(\eta_i^1-\alpha_i^1-\alpha_i^{12}-\eta_i^2+\alpha_i^2+\alpha_i^{21})\beta_i}{\sum\limits_{i=1}^{c}(\eta_i^1-\alpha_i^1-\alpha_i^{12}+\eta_i^2-\alpha_i^2-\alpha_i^{21})\sigma_i}\right), \\ F^c_\alpha(x) &= \Phi\left(\frac{\sum\limits_{i=1}^{c}(\alpha_i^1-\alpha_i^2)\beta_i}{(\alpha_i^1+\alpha_i^2)\sigma_i}\right), \\ F^{c12}_\alpha(x) &= \Phi\left(\frac{\sum\limits_{i=1}^{c}(\alpha_i^{12}-\alpha_i^{21})\beta_i}{(\alpha_i^{12}+\alpha_i^{21})\sigma_i}\right). \end{aligned} \tag{5.18}$$

6 Numerical Example

As a test network for our experiments we used a street network from the southern part of the city of Hanover (Germany), which is shown in Fig. 4, and represented by the graph in Fig. 5. We compare routes for vehicles travelling from 9 to 1.

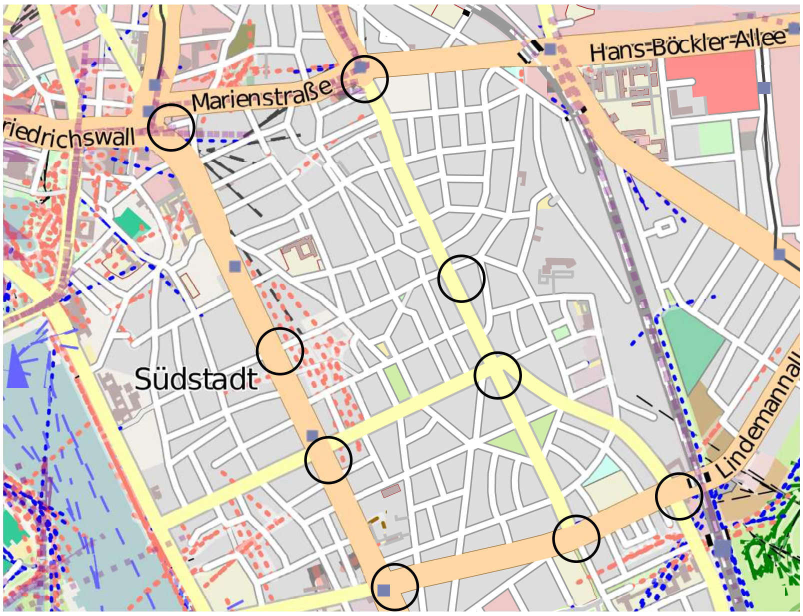

Figure 4: Test street network

We model travel times for different road segments by traffic participants (individual agents) and its aggregation by cloud agents. We assume that due to the technical or organizational limitations, travel times on different roads are indistinguishable. Travel times are collected into five samples H_1 - H_5, as demonstrated in the graph in Fig. 4.

Travel times correspond to the real-world data for this road network are represented by a mix of distributions, close to a normal one (Fig 6).

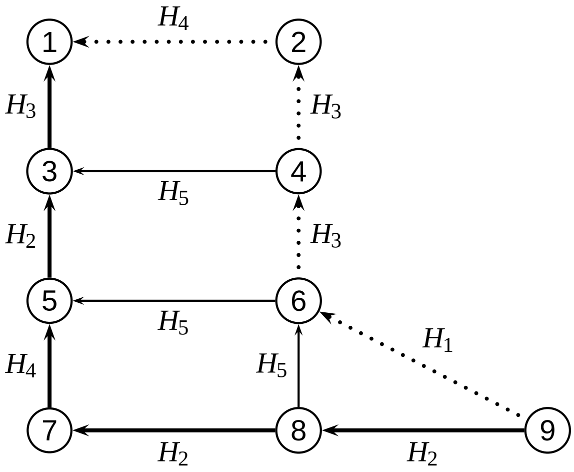

Figure 5: A graph of the street network of Fig. 4

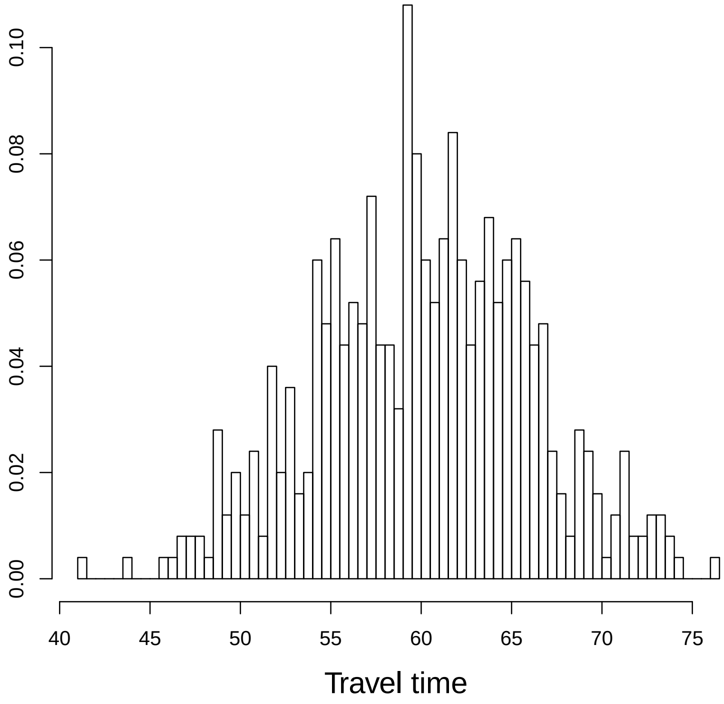

Figure 6: Histogram presenting travel times in one sample (H_1)

We used the described methods to compare and rank the routes. We applied two methods for route comparison: the resampling-based and parametric plug-in. For the resampling case, we used $N = 500$ resamples. Next, the estimated probabilities were used to construct a Markov chain and to rank the routes. The results are illustrated in Fig. 7. One can see a reliable forecast (mean is near to the exact value, dash line). Also the variance is stable, so the method provided good results for big sample sizes. One can see as well (will be illustrated later) that the resampling estimator is good for small sample sizes as well.

We compare our approach with parametric plug-in one implemented according to formula (5.15), estimating expectation β_i as $\tilde{\beta}_i$ and standard deviation σ_i as $\tilde{\sigma}_i$, replacing unknown values by their estimates. However such an approach has a number of disadvantages. First, formula (5.15) assumes normal distribution of samples. If this is not a case (the distribution is close to normal, but not exactly normal), this causes estimation error. Second, even if the distribution is normal, this estimator is biased due to the usage of the same samples for several routes. Finally, plug-in methods assume usage of all data for average and deviation calculation. If not complete information is used, this can lead to the additional bias or big variance.

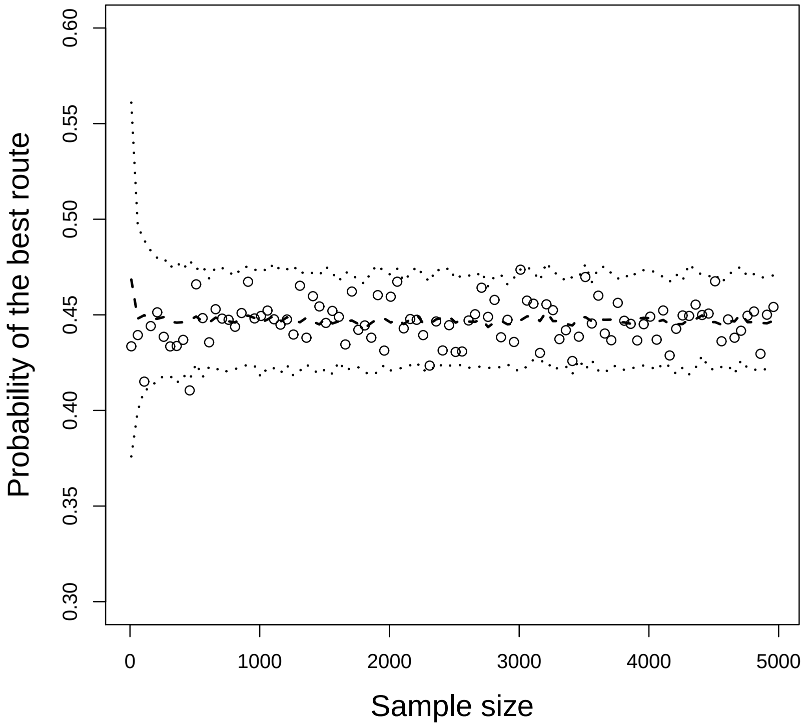

Figure 7: Resampling-based estimates of the best route probability: values (circles), true value and mean (practically identical, dashed line), deviation (dotted lines)

The results of parametric plug-in estimator are demonstrated in Fig. 8. We use only a part of available information for the calculation of mean and deviation, which is realistic working with with big data. We can see that the parametric plug-in estimator is biased, and with increase of sample size variance increases and the estimator loses precision.

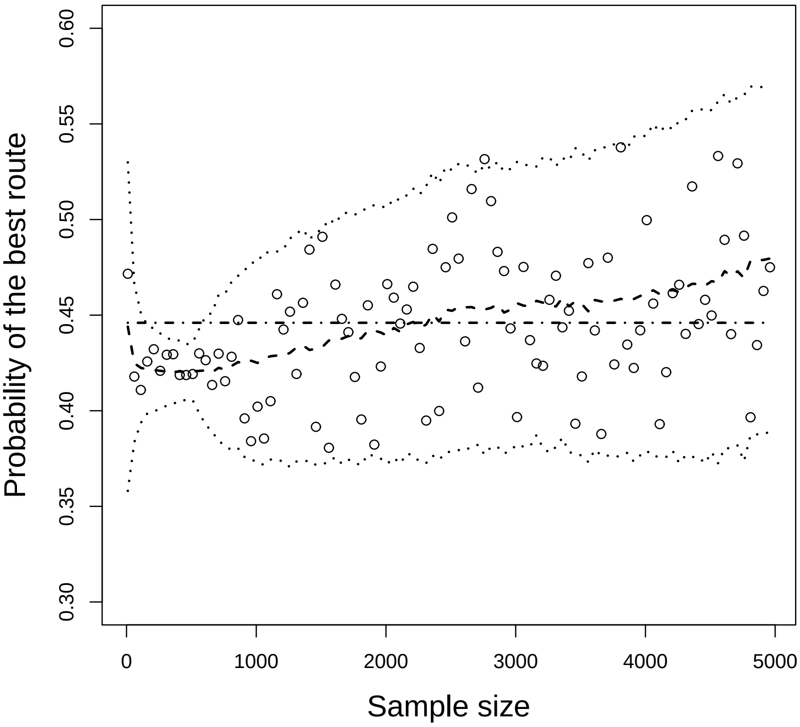

Figure 8: Plug-in estimates of the best route probability: values (circles), true value (a dash-dot line), mean (dashed line), deviation (dotted lines)

Now let us consider our theoretical results. We calculate variance of resampling and parametric plug-in estimators theoretically to compare them for small samples. For comparison, we use the mean squared errors of the plug-in $MSE(\tilde{\Theta})$ estimator and the resampling estimator $MSE(\Theta^*) = V(\Theta^*)$ because it is unbiased. The experimental results are shown in Fig. 9.
We can see that the resampling estimator is effective for small sample sizes.

Conclusion

Cloud applications open new perspectives on intelligent transportation services. Data mining is one of the most important problems for such systems. We demonstrated an approach to route recommendations in cloud-based traffic management systems. We proposed a generic cloud-based system architecture, based on the collaboration of individual and cloud agents and resampling-based route comparison approach. We applied a four-step decision process, instantiated for route recommendations and Markov chain based route ranking method for the final decision making. We investigated theoretically properties of the resampling-based route comparison and showed that it is effective alternative to parametric plug-in approach, especially for extremely small or extremely big sample sizes. Our experimental results based on real-world traffic data demonstrated advantages of the resampling-based approach, which showed more accurate results with smaller variance. Future work will be devoted to the integra-

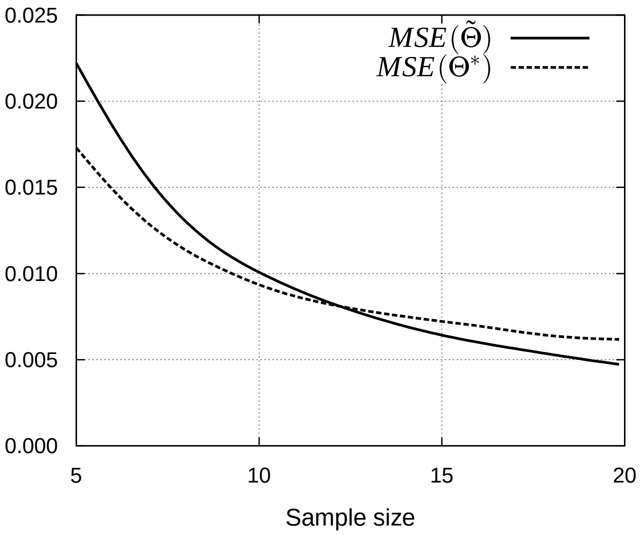

Figure 9: MSE (vertical axis) of plug-in and res. estimators for $\Theta = 0.5$

tion of the proposed algorithms to ITS and their validation on large-scale transport networks.

Acknowledgment

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.

References

Afanasyeva, H. 2005a. Resampling-approach to a task of comparison of two renewal processes, *Proc. of the 12th Int. Conf, on Analytical and Stochastic Modelling Techniques and Applications*, Riga, pp. 94–100.

Afanasyeva, H. 2005b. Resampling median estimators for linear regression model, *Transport and Telecommunication* **6**(1): 90–94.

Afanasyeva, H. and Andronov, A. 2006. On robustness of resampling estimators for linear regression models, *Communications in Dependability and Quality Management: An international Journal* **9**(1): 5–11.

Andronov, A., Fioshina, H. and Fioshin, M. 2009. Statistical estimation for a failure model with damage accumulation in a case of small samples, *Journal of Statistical Planning and Inference* **139**(5): 1685 – 1692.

Bazzan, A. L. C. and Klügl, F. 2013. A review on agent-based technology for traffic and transportation, *The Knowledge Engineering Review* **FirstView**: 1–29.

Cao, L., Luo, D. and Zhang, C. 2009. Ubiquitous intelligence in agent mining, *ADMI*, pp. 23–35.

Chen, X. 2013. *A dissertation: Analysis of Big Data by Split-and-Conquer and Penalized Regressions: New Methods and Theories*, The State University of New Jersey, New Brunswick, New Jersey, USA.

Claes, R. and Holvoet, T. 2011. Ad hoc link traversal time prediction, *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems,*, pp. 1803–1808.

da Silva, J. C., Giannella, C., Bhargava, R., Kargupta, H. and Klusch, M. 2005. Distributed data mining and agents, *Eng. Appl. of AI* **18**(7): 791–807.

Davison, A. and Hinkley, D. 1997. *Bootstrap Methods and their Application*, Cambridge university Press.

Efron, B. and Tibshirani, R. 1993. *Introduction to the Bootstrap*, Chapman & Hall.

Fioshin, M. 2000. Efficiency of resampling estimators of sequential-parallel systems reliability, *Proc. of 2nd Int. Conf. on Simulation, Gaming, Training and Business Process Reengineering in Operations*, Riga, pp. 112–117.

Fiosina, J. 2012. Decentralised regression model for intelligent forecasting in multi-agent traffic networks, *in* S. e. a. Omatu (ed.), *AISC - 9th Int. Conf. on Distributed Comp. and AI. (DCAI'12)*, Vol. 151, Springer-Verlag, Berlin Heidelberg, pp. 255–263.

Fiosina, J. and Fiosins, M. 2011. Resampling-based change point estimation, *in* J. Gama, E. Bradley and J. Hollmn (eds), *Advances in Intelligent Data Analysis X*, Vol. 7014 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 150–161.

Fiosina, J. and Fiosins, M. 2012. Distributed cooperative kernel-based forecasting in decentralized multi-agent systems for urban traffic networks, *Proc. of Ubiquitous Data Mining (UDM) Workshop of ECAI 2012*, Montpellier, France, pp. 3–7.

Fiosina, J. and Fiosins, M. 2013a. Chapter 1: Cooperative regression-based forecasting in distributed traffic networks, *in* Q. A. Memon (ed.), *Distributed Network Intelligence, Security and Applications*, CRC Press, Taylor and Francis Group, pp. 3–37.

Fiosina, J. and Fiosins, M. 2013b. Selecting the shortest itinerary in a cloud-based distributed mobility network, *in* S. O. et al. (ed.), *Proc. of 10th Int. Conf. on Distributed Computing and AI (DCAI 2013)*, Vol. 217 of *Adv. in Int. Syst. and Comp.*, Springer-Verlag, Berlin Heidelberg, pp. 103–110.

Fiosins, M. 2013. Stochastic decentralized routing of unsplittable vehicle flows using constraint optimization, *in* S. Omatu, J. Neves, J. Corchado, J. Paz and S. Gonzalez (eds), *Distributed Computing and Artificial Intelligence*, Vol. 217 of *Advances in Intelligent Systems and Computing*, Springer Berlin Heidelberg, pp. 37–44.

Fiosins, M., Fiosina, J., Müller, J. and Görmer, J. 2011a. Agent-based integrated decision making for autonomous vehicles in urban traffic, *Adv. in Int. and Soft Comp.* **88**: 173–178.

Fiosins, M., Fiosina, J., Müller, J. and Görmer, J. 2011b. Reconciling strategic and tactical decision making in agent-oriented simulation of vehicles in urban traffic, *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pp. 144–151.

Freitas, A. 2002. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer, Berlin/Heidelberg.

Gartner Reveals Top Predictions for IT Organisations and Users for 2013 and Beyond 2013. *Gartner* .

Gentle, J. E. 2002. *Elements of Computational Statistics*, Springer.

Görmer, J., Ehmke, J. F., Fiosins, M., Schmidt, D., Schumacher, H. and Tchouankem, H. 2011. Decision support for dynamic city traffic management using vehicular communication, *Proc. of 1st Int.l Conf. on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH2011)*, pp. 327–332.

Guestrin, C., Bodik, P., Thibaux, R., Paskin, M. and Madden, S. 2004. Distributed regression: an efficient framework for modeling sensor network data, *Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN '04, ACM, New York, NY, USA, pp. 1–10.

Hinneburg, A. and Gabriel, H.-H. 2007. DENCLUE 2.0: Fast clustering based on kernel density estimation, *Proc. of IDA'07, Adv. in Intelligent Data Analysis VII*, Vol. 4723 of *LNCS*, Springer-Verlag, Berlin Heidelberg, pp. 70–80.

Kleiner, A., Talwalkar, A., Sarkar, P. and Jordan, M. I. 2012. A scalable bootstrap for massive data, *Journal of the Royal Statistical Society, Series B* . (in press).

Klusch, M., Lodi, S. and Moro, G. 2003. Agent-based distributed data mining: The KDEC scheme, *AgentLink*, pp. 104–122.

Li, Z., Chen, C. and Wang, K. 2011. Cloud computing for agent-based urban transportation systems, *IEEE Intelligent Systems, IEEE Computer Society* **26**(1): 73–79.

Lin, H.-E., Zito, R. and Taylor, M. A. 2005. A review of travel-time prediction in transport and logistics, *Proc. of the Eastern Asia Society for Transportation Studies*, Vol. 5, Hamburg, pp. 1433 – 1448.

Malnati, G., Barberis, C. and Cuva, C. M. 2007. Gossip: Estimating actual travelling time using vehicle to vehicle communication, *4-th Int. Workshop on Intel. Transportation*, Hamburg.

Manolopoulou, I., Chan, C. and West, W. 2010. Selection sampling from large data sets for targeted inference in mixture models, *Bayesian Analysis* **5**: 451–464.

Negahban, S., Oh, S. and Shah, D. 2012. Iterative ranking from pair-wise comparisons., *in* P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (eds), *NIPS*, pp. 2483–2491.

Rajaraman, A. and Ullman, J. D. 2011. *Mining of Massive Datasets*, Cambridge University Press, Cambridge, UK.

Smith, B. L., Williams, B. M. and Oswaldl, R. 2002. Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies* **10**: 303–321.

Stankovic, S. S., Stankovic, M. S. and Stipanovic, D. M. 2009. Decentralized parameter estimation by consensus based stochastic approximation, *IEEE Trans. Automatic Controll* **56**.

Symeonidis, A. L. and Mitkas, P. A. 2005. *Agent Intelligence Through Data Mining*, Vol. 14 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, Springer-Verlag, New York.

Talia, D. 2011. Cloud computing and software agents: Towards cloud intelligent services, *Proc. of the 12th Workshop on Objects and Agents* **741**: 2–6.

Wu, C. 1986. Jackknife, bootstrap and other resampling methods in regression analysis, *The Annals of Statistics* **14**(3): 1261–1295.

Xiao, L. and Wang, Z. 2011. Internet of things: a new application for intelligent traffic monitoring system, *Journal of Networks* **6**: 887–894.

Zhang, C., Zhang, Z. and Cao, L. 2005. Agents and data mining: Mutual enhancement by integration, *AIS-ADM2005*, Vol. 3505 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 50–61.