Control and Cybernetics

vol. **41** (2012) No. 4

Community Traffic: a technology for the next generation car navigation*

by

K. Dembczyński[2], P. Gaweł[1], A. Jaszkiewicz[2], W. Kotłowski[2], M. Kubiak[1], R. Susmaga[2], P. Wesołek[2], A. Wojciechowski[2] and P. Zielniewicz[2]

[1] NaviExpert Sp. z o. o., Dobrzyckiego 4, 61-692 Poznań, Poland
[2] Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland

Abstract: The paper presents the NaviExpert's Community Traffic technology, an interactive, community–based car navigation system. Using data collected from its users, Community Traffic offers services unattainable to earlier systems. On the one hand, the current traffic data are used to recommend the best routes in the navigation phase, during which many potentially unpredictable traffic-delaying and traffic-jamming events, like unexpected roadworks, road accidents, or diversions, can be taken into account and thereby successfully avoided. On the other hand, a number of distinctive features, like immediate location of various traffic dangers, are offered. Using exclusively real-life data, provided by NaviExpert, the paper presents two illustrative case studies concerned with experimental evaluation of solutions to computational problems related to the community-based services offered by the system.

Keywords: community traffic, satellite car navigation, reliability analysis, travel time prediction

1. Introduction

The NaviExpert's Community Traffic (NX-CT), a crucial part of the NaviExpert Navigation System, is a technology especially designed to interact with its users. NX-CT, representing the next, more advanced generation of rapidly developing satellite-based car navigation systems, collects an assortment of data concerning the current traffic situation, which are stored, processed and finally used to recommend the best routes during the navigation phase. This means that potentially unpredictable traffic-delaying and traffic-jamming events, resulting from unexpectedly started roadworks, road accidents, closed roads or diversions, can be taken into account and thereby successfully avoided. In order to operate efficiently, the system processes massive amounts of data, which can be generally categorized into:

*Submitted: October 2012; Accepted: November 2012

- implicit data (generated automatically by the mobile application),
- explicit data (generated purposefully by the community users).

Each kind of data needs specialized procedures. For example, information generated by the users may be, for various reasons, untrue (e.g. because of being outdated). The analysis in this case involves verifying the reliability of the information sources (i.e. the reliability of those who submitted the information). Its computational challenges are illustrated in the first batch of experiments described in this paper.

At the same time, the bulk of information received by the system is used for navigational purposes, in particular for finding the fastest routes. This also calls for specialized procedures, in particular for a good travel time prediction model. The model must be fairly stable on the one hand, but flexible enough to react to the dynamically changing traffic situation on the other. Its computational challenges are illustrated in the second batch of experiments described in the paper.

This paper describes selected services offered by the NX-CT system and provides experimental illustration of the two key aspects. Following some related works described in Section 2, Section 3 presents the different generations of car navigating systems, describing their intrinsic characteristics, and introduces the NX-CT system. Sections 4 and 5 introduce two exemplary computational problems (message reliability estimation and travel time estimation, respectively) related to the community-based services offered by the system. The two sections include also two case studies concerned with experimental evaluation of those problems. The paper is concluded in the final section.

2. Some related works

A considerable amount of novelty in modern transportation system design has come from applying real–time position data collected from GPS devices installed on moving vehicles. Such dynamic approaches differ from the traditional model, in which decisions were based on statistical data collected in the past. While decision support systems built on statistics and rules form the basis for artificial intelligence, incorporating dynamic data describing current state of environment is a step towards ambient intelligence systems (e.g. Ramos et al., 2008). In this light a crucial part of modern logistic systems relies on data quality provided either anonymously (and thus automatically) or in personalized form (and thus consciously) by human operated vehicles. One intensive application area for those systems is vehicle routing. Dynamic vehicle routing problem has been discussed in numerous recent research papers and several applications have been proposed. Thus, for instance, Mukai et al. (2005) present a strategy for proactive route planning based on expected rewards, in which the learning architecture is designed to enable transport vehicles foresee the next destination. Giaglis et al. (2004) propose a generic architecture for mobile real-time systems for urban distribution, while Zografos et al. (2002) developed system to address incident response logistics.

Another intensive application area for those systems is vehicle navigation, which is also the subject of this paper. It must be noted that, apart from numerous research papers and research projects in this area, several commercial navigation solutions are already available.

For example, the systems Yanosik (`yanosik.pl`) and Coyote (`www.moncoyote.com`) offer services that include collecting user messages and utilizing these messages in danger identification procedures, while TomTom HD Traffic (`www.tomtom.com/en_gb/services/live/hd-traffic`) and Garmin 3D Traffic Live (`www.garmin.com/traffic`) offer services that include estimating travel times and utilizing these times in route finding procedures. Another example is the system Waze (`www.waze.com`), which heavily relies on the community of its users and tries to deal with both of the addressed data processing aspects. Unfortunately, comparisons of those commercial systems with the presented NX-CT system are quite difficult, as there are practically no publications on the working details of those systems, at least as far as their computational aspects are concerned.

Problems posed and solved in such systems (including NX-CT), i.e. verifying the reliability of the information sources and, first of all, predicting the travel times, were described and discussed in numerous papers, including papers on different approaches to assessing data source credibility, like Hilligoss and Rieh (2008), Kubiak (2007), Nunez et al. (2012), or Tseng and Fogg (1999), and papers on different approaches to learning prediction models from floating car data, like Billings and Yang (2006), Liu et al.(2006), Rice and van Zwet (2004), Lint et al. (2005), Wan and Kornhauser (2010), or Zhu et al. (2010).

However, community and social network based systems are being increasingly wider conceived and constructed. The objectives of these systems systematically become more general, and the use of travel related social contribution feed is no longer limited to routing or navigating. Basing on multiple-user GPS trajectories, Zheng et al. (2009) experimentally mine interesting locations and classical travel sequences in a given geo-spatial region; basing on GPS travel data, Wolf et al.(2001) try to derive the actual trip purposes; finally, basing on reports on popular travel destinations, Caschera et al.(2009) reconstruct temporal evolution of tourist targets.

3. The NaviExpert community traffic system

This section describes car navigation systems in general and the NX-CT system in particular. After outlining some earlier generations of car navigation systems, it presents the main navigating features of NX-CT. And because the idea of the system is being interactive, it enumerates the ways in which the users can interact with NX-CT. Finally, the distinctive, community-based services of the system are briefly discussed.

3.1. Generations of car navigation systems

All car navigation systems could be roughly categorized as follows:

First generation: systems capable of finding a route from A to Z through B, C, . . . , with a 'route' meaning in this context a route of minimal length. For example: find a route from Poznań to Copenhagen through Berlin. The fundamental advantage of such systems is in finding the way and navigating in unfamiliar environment. If the map data used by the system are updated on a regular basis, then the system will find routes that avoid some long-term traffic hindrances, like roadworks, closed roads, etc. The first disadvantage of such systems is that users have no control over the type of roads with which the route is created. The users who want to drive over better roads may only influence this by specifying manually a number of additional 'through' points, thus forcing particular connections, and modifying the final result. Another, fundamental disadvantage is that the system cannot find fastest routes, as the travel time basically depends on some static roadwork parameters, like type of road, but also on the dynamically changing traffic situation, including short-term traffic hindrances (especially in urban areas). Therefore, the systems are best suited for long-distance routes.

Second generation: systems capable of finding a route from A to Z through B, C, ... that satisfies some pre-defined constraints regarding selected static parameters of the road network. A 'route' still means a route of minimal length, although owing to the imposed constraints, it may be not the shortest possible one. For example: find a route from Poznań to Kraków by motorways (wherever possible). Systems of this kind extend their functionality to finding shortest routes of at least given quality (if available). This gives the users better control over the connection, as types of roads selected for the route influence both travel time and travel comfort (travelling along roads of better quality may shorten the average travel time and help maintaining the driver's confidence). The remaining disadvantage is the system's inability to find fastest routes and to react to the current traffic situation, including short-term hindrances, etc. The systems are suitable for long-distance and medium-distance routes.

Third generation: systems capable of finding both shortest and fastest routes from A to Z through B, C, ... additionally satisfying pre–defined constraints regarding static parameters of the road network. The shortest 'route' means the one of minimal length, while the fastest 'route' means the one of minimal travel time, although combined routes can also be successfully defined and found. For example: find the fastest route from Poznań coach station (located close to the city centre) to Poznań airport (located a few kilometres west of the city) about noon. In finding fastest routes these systems can use dynamically collected traffic data, e.g. floating car data, to find connections of greater average speed, avoiding regularly congested areas and usual traffic jams. The final disadvantage is that the system is incapable of reacting to short-term traffic hindrances and obstructions, including sudden jams or detours caused by road accidents. The systems are generally suitable for all-distance routes (long, medium and short), including traffic-troubled, congested urban areas.

Fourth generation: interactive systems, capable of reacting to long-term as well as short-term traffic situations when finding both shortest and fastest routes from A to Z through B, C, ... that additionally satisfy pre-defined constraints regarding static parameters of the road network. Again, shortest, fastest as well as combined routes can be found. The systems are fully operable for all-distance routes (long, medium and short) and for all-area environments. The development of such systems is directed towards building community networks of their users. Owing to the interactive manner of their operation, the systems are fully dynamic, and thereby capable of avoiding unpredictable traffic-delaying and traffic-jamming events, resulting from unexpectedly started roadworks, road accidents, closed roads or detours.

3.2. Next generation car navigation in NX-CT

Previous systems essentially lacked the fully dynamic functionality, because they possessed no direct information on unpredictable traffic events. Admittedly, they were able to react to many of these situations, but – owing to their mode of operation – their reaction could only come after a considerable delay. This is because the dynamic information that these systems collect and utilize consists of the floating car data, that is time-stamped geographical positions of the GPS devices (and thus the vehicles that carry them). These raw positions are converted to passages through road segments of the underlying road network, which, under proper assumptions, permits the system to draw more or less accurate conclusions regarding the general fluency of the traffic on particular road segments. The most immediate deductions regard the actual average speeds of the passages: if the average speed of all passages observed at a given time on a given segment decreases considerably, then this probably signifies a traffic jam there. The ultimate conclusion is then that passing this segment in immediate future will also follow that decreased (or at least not significantly different) speed. In result, it is advisable to avoid the particular segment when planning fast routes in favour of other segments, which may make the route longer, but ultimately faster.

The floating car data, i.e. time-stamped geographical positions of the GPS devices, constitute implicit data supplied to the system. However, there exist also an non-implicit source of data that can assist the system, especially in searching for fastest routes. The main difficulty that faces the fastest route planning systems of previous generations is the lack of observational data. Consider a road segment for which no passages have been observed for some recent time. This may be due to chance (no user of the system drove just there), with the traffic being fluent, so redirecting cars through this segment makes good sense. Unfortunately, observing no passages through a given segment may also imply that owing to some unpredictable traffic situation (e.g. a serious road accident) the segment had been entirely closed for traffic. In this case, redirecting cars through this segment makes no sense.

An interesting solution to this problem is achieved by inviting the system's users to participate in the process of generating the necessary data. Because NX-CT is a mobile phone-based system, run as a mobile phone-based application, its users have direct possibility of engaging into active interaction with the system. First of all, by generating and submitting an appropriate message, the user can notify the system (and thereby its whole user community) about a specific traffic situation, like new detours being just assembled or old ones disassembled, new road segments being just opened or old ones being just closed, etc.

The notification data, i.e. messages and alerts submitted by the community, constitute explicit data supplied to the system. The notification data may found diverse, more or less direct, applications. A direct application is in current route finding. A less direct application may involve creating maps, plans or other depictions, that can be used by all users of the community in various studies, reports and accounts.

It should be also finally stressed that all the community users, apart from providing various forms of information (implicit and explicit) to the system, obtain explicit information from the system. This information permits the users to find fastest routes, especially in response to rapidly changing traffic situation and thereby to travel more safely and more comfortably.

3.3. Ways of interacting with NX-CT

The users can interact with the system by submitting messages, which they can do at any time – first of all, from their cars, but also from outside the cars. To this end, they have a number of interfaces at their disposal, which include the mobile application, a web-based internet portal, a web-based internet forum, the e-mail, and the phone. All these interfaces have their intrinsic characteristics, for example the internet forum gives the user the immediate opportunity to discuss an issue with other users interested in the subject.

The submitted messages can generally be categorized into confirming and cancelling messages. A great deal of various situations can be reported, including traffic-specific: closing/opening of road segments, commencement/completion of roadworks, changes/updates to road infrastructure and to road signs, errors/omissions in road maps, as well as many other situations, including: road dangers, road checks, speed cameras.

For various reasons, the different pieces of information submitted by the users may be untrue (for example because they are no longer up-to-date). This is why the systems attempts to verify the received messages. Verifying such kind of information is, in general, a complex problem. The idea actually utilized here is that of verifying the reliability of the information sources (i.e. of those users who submitted the particular pieces of information).

3.4. Distinctive services of NX-CT

In addition to route finding and navigating, the NX-CT system offers its users numerous other services. Apart from quite typical, map-oriented functionality, like calculating parameters of various routes (including routes found by the system) or displaying various map layers (including the points of interest layer), the system provides the following, specialized services:

- characterizing and visualizing the current traffic state of selected areas in real time,
- finding approximate geographical position for devices without the GPS functionality (so called 'cell ID' identification).

Especially interesting services of the system arise from cooperation with other communities. This includes utilizing evaluations of some pre-defined objects (e.g. points of interest) supplied by users of those communities. For example, the system may recommend:

- restaurants, evaluated by the users of gastronauci.pl,
- natural/architectural monuments, evaluated by the users of wikipedia.pl.

Finally, the system's community can also influence many very system-specific issues. Among others, these may include suggestions concerning:

- preferences on recommended routes (i.e. routes that should be offered as first to users),
- road categorization (i.e. the quality classes to which all roads are conventionally assigned),
- navigating messages (i.e. the messages voiced by the system in the navigating phase),
- points of interest (i.e. the locations that may turn out to be really worth visiting).

4. Estimating the reliability of messages

This section illustrates various analyses of the explicit warning reports against road dangers, speed cameras, or road checks, submitted to the system by the community users. Unfortunately, such submissions are often quite scattered as far as their location is concerned, because different users move at different speeds in different directions and, additionally, they generate their messages with different delays. In result, locations of warnings that concern the same event may vary considerably. To be useful, however, these reports should be not only true but also as accurate as possible, at least as far as the locations reported in them are concerned. Their analysis is therefore twofold. Firstly, the reports are clustered to discover distinct events and, secondly, their reliability is verified. Below, we illustrate the second phase of the analyses.

4.1. Modified voting

The simplest idea of computing the reliability of a warning against an event involves computing the ratio of positive reports (i.e. messages that confirm the existence of the event) to all reports, the procedure referred to as ‘voting’. Let n be the number of all reports in a group of reports and pos the number of positive reports in the same group. Then the voting reliability of a warning is equal to $\frac{pos}{n}$.

This voting approach may be slightly modified in order to reduce the reliability of warnings characterized with only few reports: one may notice that when there is only one positive report in a group, then the generated warning would receive reliability of 100%. Therefore, the modified voting reliability is computed as $\frac{pos}{n} \times \frac{n+1}{n+2}$.

4.2. Expectation maximization

Another idea involves building a specialized probability model for the given data generation scenario. All the variables involved in the scenario are binary: a report is either positive or negative, a warning either exists or not. Thus, a probabilistic model is not difficult to establish (see Kubiak, 2007).

Let n_e and n_u be the number of events and users, respectively. Each user u_i, $i = 1, \ldots, n_u$, may send a report concerning an event e_j, $j = 1, \ldots, n_e$. Let us further assume that we have a set D of such reports represented by binary variables r_{ij}, stating whether a user u_i confirmed or did not confirm the event e_j. The probability of the observed data can be then expressed by:

$$p(D) = \prod_{(i,j)\in D} \left(p(u_i)\left[p(e_j)^{r_{ij}}(1-p(e_j))^{1-r_{ij}}\right] + (1-p(u_i))\left[(1-p(e_j))^{r_{ij}} p(e_j)^{1-r_{ij}}\right] \right),$$

where $p(e_j)$ is probability of a positive event e_j (i.e., the reliability of a warning) and $p(u_i)$ is probability that a user u_i sends a reliable report (i.e, the reliability of the user). Although these parameters are initially unknown, their values may be estimated using the submitted reports. The problem can be formulated and solved by maximizing the likelihood of observed data, $p(D)$, which is the core of the Expectation Maximization algorithm (see Dempster et al., 1977).

4.3. Experimental study

The two methods of reliability estimation were compared on a set of user reports generated during a nine-month period of 2007 in the area surrounding the city of Poznań. Only reports related to speed cameras were used; 954 reports were available in this setting.

The modified voting and the EM algorithm both use a reliability threshold to filter unreliable warnings. The values of the threshold were varied from 0 to 1 with step of 0.2.

A reference set of warnings (the so-called ground-truth) was available in this experiment, as precise information on the existence of speed cameras in the 43 places mentioned in users’ reports was acquired. In 29 cases the speed cameras did exist (reliability equal to 100%), while in 14 cases they did not exist

Table 1. Reliability estimation results of the two methods

Threshold	Voting		EM	
	#warnings	MSE	#warnings	MSE
0.0	34	0.120	35	0.094
0.2	34	0.120	31	0.080
0.4	32	0.118	22	0.022
0.6	29	0.109	20	0.000
0.8	16	0.013	19	0.000
1.0	00	0.000	17	0.000

(reliability equal to 0%). One may notice, however, that the reference set is not a proper random sample of potential speed camera positions.

To measure the quality of the approaches we use the number of warnings reported by the methods and the mean square error (MSE) of the reliability of the reported warnings with respect to the ground-truth. We only consider warnings that matched the ground truth (this is an optimistic estimate, as we do not count warnings not related to any of the considered 43 potential places).

The results of the experiment are shown in Table 1. Its contents reveals that the EM algorithm significantly outperforms the voting method for all thresholds. It is also worth noting that, starting from the threshold equal to 0.6, the EM algorithm generates 20 ground-truth warnings with perfect precision: MSE series approaches 0. A similar case for the voting algorithm starts from 0.8, but then the number of matched warnings starts to fall and its drop in MSE is mainly due to that fall.

5. Estimating the travel time

This section illustrates the analysis of data for finding fastest routes, which can be effectively found only when the system has access to accurate estimates of travel times for each road segment. In other words, the goal is to predict the vehicle travel time between two given points on a road network, which, in order to reduce its computational complexity, is cast to that of estimating the travel time on single road segments. The analysis is entirely based on the implicit data, in the form of time-stamped GPS positions, sent automatically from the users of the traffic network.

5.1. The prediction model

More formally, the goal of the problem can be stated as prediction of unknown value of vehicle travel time y_{st} on a particular road segment $s \in \{1, \ldots, S\}$ in a given time point t. The task is then to find a function $f(s, t)$ that estimates, in the best possible way, the value of y_{st}. The accuracy of a single prediction $\hat{y}_{st} = f(s,t)$ is measured by a loss function $L(y_{st}, \hat{y}_{st})$, which determines the penalty for predicting $\hat{y}_{st}$ when the true value is y_{st}. A reasonable loss function in this case is the squared error loss:

$$L(y_{st}, \hat{y}_{st}) = (y_{st} - \hat{y}_{st})^2.$$

Ideally, we would like to get a model $f(s,t)$ that minimizes the *expected* risk:

$$f(s,t)^* = \arg\min_f R(f) = \arg\min_f E_{(s,t)} E_{y|(s,t)} \left[(y - f(s,t))^2 \right].$$

Since this is directly impossible, as the distribution of y given (s,t) is hardly ever known, we rely on a set of training samples, $\{(y_i, s_i, t_i)\}_{i=1}^N$, and construct a model that, instead, minimizes the *empirical* risk:

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, f(s^{(i)}, t^{(i)})),$$

possibly with a kind of regularization over the function f to prevent overfitting of the model (see Hastie et al., 2003).

We additionally assume that for each road segment s and time point t, a vector $\mathbf{x}_{st} = (x_{st1}, x_{st2}, \ldots, x_{stn})$ of attributes, which describe the segment and the time point, is known. Without the loss of generality, we assume that attribute values are real numbers, i.e. $\mathbf{x} \in \mathcal{R}^n$.

5.2. The static and dynamic components

The whole procedure comprises two models. The first model, the static one, is responsible for predicting overall trends in the traffic. It assumes that the traffic undergoes periodic changes, but is otherwise static. The model is developed on the basis of a set of past observations, discovering (potentially existing in the data) repeatable traffic flow patterns (e.g. "at every Sunday morning, on a road segment in the city center, the traffic is low"). This constitutes its strength (the stability of predictions, ensured by large data samples and the ability to predict for the long-term, e.g. with a horizon of a few days), but also its weakness (the inability to react to dynamically changing, non-periodic traffic conditions). This poor reactivity is also the reason for introducing the second model, the dynamic one, which exploits the most recent, real-time observations with the aim to improve the short-term predictions of the static model.

5.2.1. The static component

Construction of the static model is similar to the typical regression task. To deliver the right prediction, the model uses attributes that describe a given segment at a given time point. The training data are represented in a tabular form $\{(y_i, \mathbf{x}_i)\}_{i=1}^N$. In this study we use rather simple static models that exploit only a limited number of features describing a road segment and a time point.

However, the models described here behave fairly satisfactorily and for the sake of readability we limit our discussion to them.

The simplest method for predicting the travel time y relies on estimating a single value from all observations. Such a value corresponds to the average unit travel time for the considered road network and the time interval. More precisely, we compute the average inverse velocity (the average travel time for a length unit) over all historical observations:

$$\bar{v}^{-1} = \frac{\sum_{i=1}^{N} y^{(i)}}{\sum_{i=1}^{N} x_l^{(i)}},$$

where x_l is the length of the l-th segment. The prediction for a given road segment is then given by:

$$\hat{y}_{gm} = f_s(\mathbf{x}) = x_l \times \bar{v}^{-1}.$$

This form of the prediction is reasonable, as the average inverse velocity is the solution to the optimization problem:

$$\bar{v}^{-1} = \arg\min_a \sum_{i=1}^{N} x_l^{(i)} \left(\frac{y^{(i)}}{x_l^{(i)}} - a \right)^2,$$

where the length of the segment is multiplied by the loss for single observations. In other words, if we minimize the weighted squared loss on a training set, the average inverse velocity is the best possible choice for the estimated single value. We refer to this model as the *global mean*.

The second method, referred to as the *segment mean*, averages the travel time on each road segment separately. Although more specific than the global mean, it is still primitive enough to ignore the time point of the passage:

$$\hat{y}_{sm} = f_s(\mathbf{x}) = \frac{\sum_{\mathbf{i}:\mathbf{x}_{\mathbf{id}}^{(\mathbf{i})}=\mathbf{x}_{\mathbf{id}}} \mathbf{y}^{(\mathbf{i})}}{\sum_{\mathbf{i}:\mathbf{x}_{\mathbf{id}}^{(\mathbf{i})}=\mathbf{x}_{\mathbf{id}}} \mathbf{1}}.$$

The third model, referred to as the *segment/time period mean*, additionally considers information about the time point of the passage. Using expert knowledge on traffic trends, we define five time periods and separately compute the segment mean for each roach segment in each time period.

5.2.2. The dynamic component and the higher level combination

The goal of the dynamic model is to use the most recent observations to improve the predictions of the static model $f_s(\mathbf{x}_{st})$ in the short-term. The dynamic model is introduced to account for those changes in traffic that cannot be explained by exploiting long-term and periodic behaviour.

The dynamic model f_d is structured hierarchically, with a static and a dynamic component at its lower level. The dynamic component is constructed as a time series model for each road segment. Prediction $\hat{y}_{st_0}$ for a given segment s

and time point t_0 is computed using previous observations y_{st}, $t < t_0$, from segment s. Training data are represented in form $(y_{st_1}, y_{st_2}, \ldots, y_{st_{N_s}})$, for each segment $s \in \{1, \ldots S\}$, where N_s is the number of observations for s. In this paper we use a simple moving average over all past observations from a given time interval t:

$$f_{ma}(s,t) = \hat{y}_{st_0} = \frac{\sum_{t_0 - t_i < t} y_{st_i}}{\sum_{t_0 - t_i < t} 1}.$$

At the higher level the model takes as input the prediction from the static model $f_s(\mathbf{x}_{st})$, the prediction from the moving average model $f_{ma}(s,t)$, and produces the final travel time estimate as a linear combination of the segment length and the two previous predictions:

$$f_d(s,t) = a_0 x_l + a_1 f_s(\mathbf{x}_{st}) + a_2 f_{ma}(s,t).$$

This model is thus a kind of a cascade, in which the static prediction is combined with the dynamic moving average. The model is trained by linear regression every 5 minutes on the most recent observations from time window of a few past hours. The coefficient a_0 controls the change of the average travel time per unit length, the coefficient a_1 is responsible for the proportional adjustment of the static model to the current traffic, while the coefficient a_2 determines the reliability of the prediction computed from the most recent observations.

5.3. Experimental study

In our experiments we use real-life floating car data provided by NaviExpert. All the data were collected from a pre-defined geographical area in a pre-defined time range. The area of observation ranges from 16.94190° N to 16.95980° N and from 52.39294° E to 52.41417° E, covering nearly 3 square kilometres in the city of Poznań. The area contains two important roundabouts: Rondo Śródka and Rondo Rataje, with Jana Pawła II Street between them, as well as two important bridges: Most Bolesława Chrobrego and Most Św. Rocha. This particular area was chosen because in 2011, on 26th of September, it was affected by a specific incident that we study in greater detail in the experiment.

The time range of the observations spans four weeks in 2011: since 12th Sept till 10th Oct, with the exception of the night hours (from midnight until 5 a.m.).

In total, we use four methods for travel time estimation: the global mean (GM), the segment mean (SM), the segment/time period mean (STPM), and the dynamic model (DM). For computing linear regression we use the Weka package (see Hall et al., 2009).

To build the dynamic model we take the most recent observations from a time window of exactly one hour.

Table 2. Prediction results of the four models

model	MAE[min]	MAE[%]	RMSE[min]	RMSE[%]
Global Mean	0.3307	119.80	0.8556	108.00
Segment Mean	0.2761	100.00	0.7922	100.00
Seg./Time Period Mean	0.2649	95.97	0.7776	98.15
Dynamic model	0.2556	92.60	0.6415	80.98

5.3.1. General performance

The first experiment evaluates the general performance of the models. To this end, we split the data into learning and testing parts: the learning set extends between 12th and 25th September, while the test set extends between 26th September and 10th Oct. The static models are built using only the learning set, while the dynamic model additionally uses the most recent observations from the test set (but each prediction is entirely based on earlier observations). The performance of all models is presented in Table 2. We report both mean absolute error (MAE) and root mean squared error (RMSE), with the result of the segment mean as the reference for computing the relative errors.

As it can be observed, SM and STPM improve significantly over GM, although it is DM that achieves the best results, particularly in terms of RMSE. This is due to the adaptive nature of this model.

5.3.2. Specific performance

The second experiment concerns a specific traffic situation. It focuses on an accident that occurred on 26th September, 2011 in the selected area. The local press reported* that day to be generally affected by traffic jams in the whole city, but specifically in the selected area: a lorry broke down in the Jana Pawła II Street and was removed only about 9 p.m., resulting in unusual congestion lasting until late evening hours. The accident coincided with the beginning of a new academic year, which additionally increased the traffic.

Actual tests concern the performance of the models on two separate days, 19th and 26th of September. The former was chosen to precede the latter by exactly one week, to be used for reference and comparison with the fairly unusual 26th September. For each of them a learning set spanning a week before the chosen day is extracted and the static models are constructed using these data sets.

Fig. 1 shows the RMSE for the static time period model and the dynamic model throughout the two compared days. On 19th September the prediction errors of the static and the dynamic models are similar and fairly low, as no

*`http://poznan.gazeta.pl/poznan/1,36037,10359810,Poznan_sparalizowany__Wina_jednego_tira_.html` ("Poznań jammed, one lorry to blame"), in Polish.

Figure 1. Prediction error for models by the time of day, for two different days: boxes SM 26th Sept., diamonds DM 26th Sept., triangles SM 19th Sept., circles DM 19th Sept.

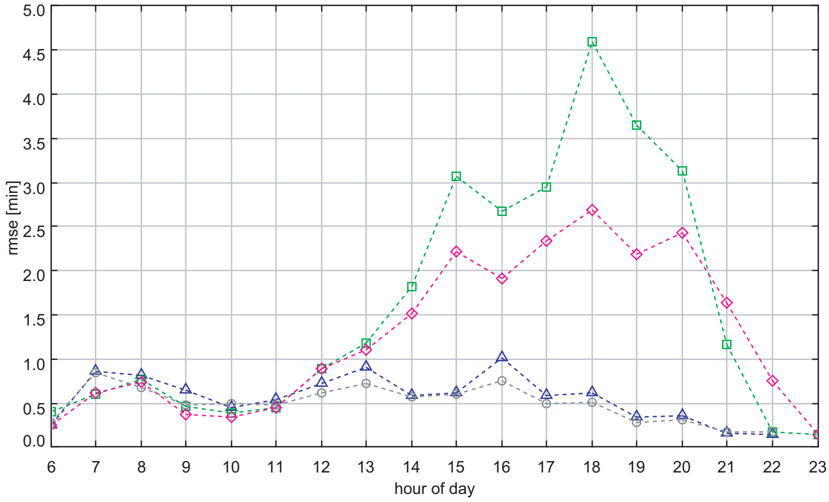
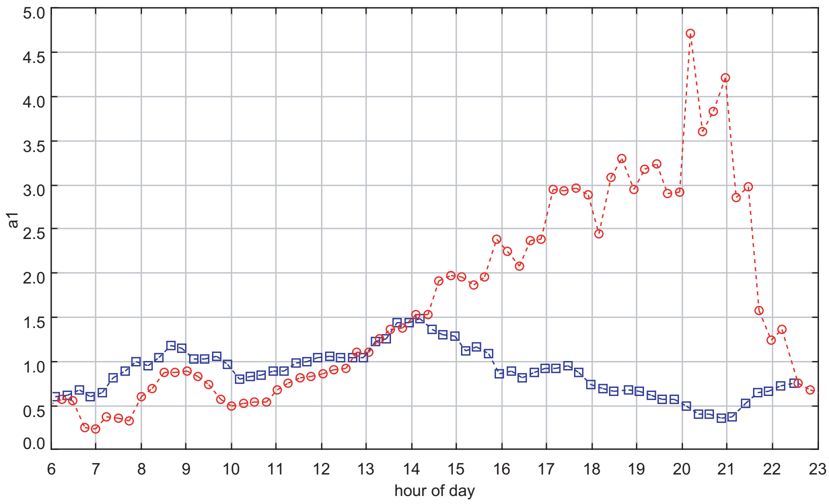


Figure 2. Regression coefficient a_1 of the dynamic model by the time of day, for two different days: circles 26th Sept., boxes 19th Sept.



major incident happened on that day, and the traffic conditions were typical, producing quite accurate predictions of the static model and a slight, but consistent improvement the dynamic model over the static one.

On the other hand, the prediction error of both models rises significantly in the afternoon of 26th September. As far as this error is concerned, the dynamic model outperforms the static model, as is able to adapt to the unexpected accident. According to the local press, the broken lorry was removed about 9 p.m., which is visible in the figure, as the prediction error starts to fall down around this hour. The predictions of the dynamic model are a bit worse after 9 p.m., which may be caused by the (decreasing but existent) bias of the previous events that were still present in the one-hour time window of the model.

The behaviour of the dynamic model is reflected by the values of the a_1 coefficient, which adapts the static model to the current traffic situation. Values greater than 1 suggest the presence of unusual traffic jams, while values lower than or equal to 1 suggest usual traffic flow. As it can be seen in Fig. 2, the afternoon values of a_1 differ significantly between the two days under consideration. On 19th September, a_1 slightly oscillates around the default value of 1, suggesting that the dynamic traffic conditions match the static, historical pattern. On 26th September, a_1 starts rising in the afternoon, approaching 5 at about 8 p.m., which reflects the extreme and unusual increase of the travel time in the area, lasting well into the evening hours. After 9 p.m., as the incident has been dealt with, the values of a_1 fall rapidly, suggesting return of the normal traffic conditions.

6. Conclusions

The paper describes the range of services offered by the NaviExpert's Community Traffic system, a next generation interactive technology that uses various kinds of user-supplied data for finding and recommending best routes during the navigation phase. The development of such systems is directed towards building community networks of their users. Interacting actively with the system, the community can provide data of enormous usability. Their most obvious application is in current route finding, which in result becomes much more reactive to unpredictable traffic-delaying and traffic-jamming events. Another, exclusively community-oriented, application is in shaping the system services, the quality of which may be positively influenced by the community's feedback. Still another application, arising from cooperation with other communities, includes utilizing evaluations of pre-defined objects (e.g. points of interest) supplied by users of those communities.

In two small case studies the paper illustrates an experimental evaluation of two important aspects of the complex data processing carried out by the system: the reliability of information submitted by the community, and the flexibility of the travel time prediction. In each case, two different types of methods were tested: a basically simple, but computationally little demanding method (simple voting in reliability estimation and simple averaging in travel time estimation) and a more advanced, but computationally more demanding method (expectation maximization in reliability estimation and dynamic model in travel time estimation). In both cases the more advanced methods significantly outperformed the simple ones,

achieving results that make these methods useful enough to be used in practical applications, despite their increased computational demands.

Acknowledgement

The research has been a part of the UDA-POIG.01.04.00-30-066/11-00 project, carried out by NaviExpert Sp. z o. o., co-financed by the European Regional Development Fund under the Operational Programme 'Innovative Economy'.

References

Billings, D. and Yang, J. (2006) Application of the ARIMA models to urban roadway travel time prediction — a case study. *IEEE International Conference on Systems, Man and Cybernetics* **3**, 2529–2534.

Caschera, M.C., Ferri, F., Grifoni, P. and Guzzo, T. (2009) Multidimensional visualization system for travel social networks. *Proceedings of the 6th International Conference on Information Technology: New Generations* IEEE Computer Society Washington, DC, 1510–1516.

Dempster, A., Laird, N. and Rubin, D. (1997) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **39**, 1, 1–38.

Giaglis, G., Minis, I., Tatarakis, A. and Zeimpekis, V. (2004) Minimizing logistics risk through real-time vehicle routing and mobile technologies: Research to date and future trends. *International Journal of Physical Distribution and Logistics Management* **34**, 9, 749–764.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009) The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* **11**, 1, 10–18.

Hastie, T., Tibshirani, R. and Friedman, J.H. (2003) *Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag.

Hilligoss, B. and Rieh, S.Y. (2008) Developing a unifying framework of credibility assessment: Construct, heuristics, and interaction in context. *Information Processing and Management* **44**, 1467–1484.

Kubiak, M. (2007) Credibility assessment in an on-line car navigation system by means of the Expectation Maximization algorithm. *Foundations of Computing and Decision Sciences* **32**, 4, 275–294.

Liu, H., van Lint, H., van Zuylen, H. and Zhang, K. (2006) Two distinct ways of using Kalman filters to predict urban arterial travel time. *Proceedings of IEEE Intelligent Transportation Systems Conference.* IEEE Toronto, 845–850.

Mukai, N., Watanabe, T. and Feng, J. (2005) Proactive route planning based on expected rewards for transport systems. *Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence.* Hong Kong, 51–57.

Nunez, R., Wickramarathne, T., Premaratne, K., Murthi, M., Kuebler, S., Scheutz, M. and Pravia, M. (2012) Credibility assessment and inference for fusion of hard and soft information. *Advances in Design for Cross-Cultural Activities, Part I.* CRC Press, 96–105.

Ramos, C., Augusto, J.C. and Shapiro, D. (2008) Ambient intelligence: the next step for artificial intelligence. *IEEE Intelligent Systems* **23**, 2, 15–18.

Rice, J. and van Zwet, E. (2004) A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems* **5**, 3, 200–207.

Tseng, S. and Fogg, B.J. (1999) Credibility and computing technology. *Communications of the ACM* **42**, 5, 39–44.

van Lint, J., Hoogendoorn, S. and van Zuylen, H. (2005) Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research, Part C* **13**, 5–6, 347–369.

Wan, K. and Kornhauser, A. (2010) Turn-by-turn routing decision based on copula travel time estimation with observable floating-car data. *Transportation Research Board 89th Annual Meeting* Paper # 10-2723.

Wolf, J., Guensler, R. and Bachman, W. (2001) Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board* **1768**, 125–134.

Zheng, Y., Zhang, L., Xie, X. and Ma, W.Y. (2009) Mining interesting locations and travel sequences from gps trajectories. *Proceedings of the 18th International Conference on World Wide Web* ACM New York, NY, 791–800.

Zhu, T., Kong, X., Lv, W., Zhang, Y. and Du, B. (2010) Travel time prediction for float car system based on time series. *Proceedings of the 12th International Conference on Advanced Communication Technology* **2**, 1503–1508.

Zografos, K.G., Androutsopoulos K.N., and Vasilakis, G.M. (2002) A real-time decision support system for roadway network incident response logistics. *Transportation Research, Part C* **10**, 1–18.