Community Traffic: A Technology for the Next Generation Car Navigation

Przemysław Gaweł, Krzysztof Dembczyński, Wojciech Kotłowski,
Marek Kubiak, Robert Susmaga, Przemysław Wesołek, Piotr Zielniewicz,
and Andrzej Jaszkiewicz

Abstract The paper presents the NaviExpert's Community Traffic (CT) technology, an interactive, community-based car navigation system. Using data collected from its users, CT offers services unattainable to earlier systems. On 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, closed roads or diversions, can be taken into account and thereby successfully avoided. On the other hand, a number of distinctive features, like immediate localization 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.

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

1 Introduction

The Community Traffic (CT), a crucial part of the NaviExpert Navigation System, is a technology especially designed to interact with its users. CT, representing the next, more advanced generation of rapidly developing satellite-based car navigation systems, collects an assortment of data concerning the

P. Gaweł (pgawel@naviexpert.pl), M. Kubiak
NaviExpert Sp. z o. o., Dobrzyckiego 4, 61-692 Poznań, Poland

K. Dembczyński, W. Kotłowski, R. Susmaga, P. Wesołek, P. Zielniewicz, A. Jaszkiewicz
Institute of Computing Science, Poznań University of Technology,
Piotrowo 2, 60-965 Poznań, Poland

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 implicit data (automatically generated by the mobile application) and explicit data (generated purposefully by the community users). Each kind of data needs specialized procedures. For example, the 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 the 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.

Several other commercial navigation solutions exist with similar purpose. 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.

Problems posed and solved in such systems (including the CT system), 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 [3, 4, 6, 8] and papers on different approaches to learning prediction models from floating car data [1, 5, 7, 9, 10, 11].

This paper describes selected services offered by the CT system and provides experimental illustration of the two key aspects. Following this introduction, Section 2 presents the different generations of car navigating systems, describing their intrinsic characteristics, while Sections 3 and 4 introduce two exemplary computational problems 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 Navigation Systems

Early navigation systems essentially lacked the functionality of collecting data from their users and reacting to the dynamically changing traffic situation. In these systems, the route finding was based on information stored within the system, with fairly limited updating capabilities.

CT uses a new car navigation technology, one which relies on bidirectional communication between the system and its users. Being a mobile phone-based application, it allows the users to engage into active interaction with the system. In general, one can distinguish two kinds of data exchange in CT: implicit and explicit.

Car floating data, i.e., time-stamped geographical positions of the GPS devices (and thus the vehicles that carry them), are collected and sent to the system implicitly. These raw positions are converted to passages through road segments (i.e. road units between two adjacent junctions) 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 the segments. The most immediate deductions regard the actual average speeds of the passages. In result, when searching for fastest routes the system may find it advisable to avoid a particular segment in favour of other segments, which may make the route longer, but ultimately faster.

The remaining difficulty in the fastest route planning is the lack of data on passages through segments. Consider a road segment through which no passages have been observed for some recent time. This may imply that there is no traffic there, so redirecting cars to 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.

To deal with this problem, CT allows its users to generate and submit appropriate messages that inform the system (and thereby its whole user community) about specific traffic situations, like new diversions, various road dangers, speed cameras, etc. The submitted messages can generally be categorized into reporting (or confirming) messages and cancelling messages. 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).

In addition to route finding, navigating, and gathering user reports, the CT system offers numerous other services, like characterizing and visualizing the current traffic state of selected areas in real time or finding approximate geographical position for mobile phones not equipped with GPS functionality (so-called ‘cell ID’ identification). Some especially interesting services

arise from cooperation with other communities and involve utilizing recommendations supplied by users of those communities, e.g. recommendations of restaurants, supplied by the users of `gastronauci.pl` or recommendations of natural/architectural monuments, supplied by the users of `wikipedia.pl`. Finally, the system's community can also influence many very system-specific issues, like road categorization or navigational messages.

3 Estimating the Reliability of Submitted Messages

This section illustrates the analysis of warning reports against road dangers, speed cameras, and 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 in different directions and, additionally, they generate their messages with various 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 as far as their locations 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.

3.1 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 report/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 this 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 $\frac{pos}{n} \times \frac{n+1}{n+2}$.

3.2 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 it does not. Thus, a probabilistic model is not difficult to establish [4].

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 a probability of a positive event e_j (i.e., the reliability of a warning) and $p(u_i)$ is a 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 (EM) algorithm [2].

3.3 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 0.1 step.

A reference set of warnings (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 (reliability equal to 0%). One may notice, however, that the reference set is not a properly drawn 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 makes it an optimistic estimate, as we do not count warnings that are 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

Table 1 Comparison of the two methods for estimating reliability of warnings: the voting method and the EM algorithm

Threshold	Voting		EM	
	#warnings	MSE	#warnings	MSE
0.0	34	0.120	35	0.094
0.1	34	0.120	34	0.096
0.2	34	0.120	31	0.080
0.3	34	0.120	31	0.080
0.4	32	0.118	22	0.022
0.5	30	0.115	21	0.014
0.6	29	0.109	20	0.000
0.7	24	0.052	20	0.000
0.8	16	0.013	19	0.000
0.9	09	0.006	19	0.000
1.0	00	0.000	17	0.000

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.

4 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's 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.

4.1 The Prediction Model

More formally, we formulate the problem as a prediction of an unknown value of the 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 the value of y_{st} using a set of training samples $\{(y_i, s_i, t_i)\}_{i=1}^{N}$. We measure the accuracy of a single prediction $\hat{y}_{st} = f(s,t)$ by a loss function $L(y_{st}, \hat{y}_{st})$, which determines the penalty for predicting $\hat{y}_{st}$ when its true value is y_{st}. A reasonable loss function in this case is the squared error loss:

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

The whole procedure involves constructing two distinct models, which are finally merged into one, combined model.

The first model, referred to as static, is responsible for predicting overall trends in the traffic. It uses 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 centre, the traffic is low"). This stability constitutes its strength (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 traffic situation).

This poor reactivity is the main reason for introducing the second model, referred to as dynamic, which exploits recent observations in real-time. Its goal is to use the most recent of the incoming data to improve the short-term predictions of the static model $f_s(s,t)$. The dynamic model is introduced to account for those changes in the traffic that cannot be explained by exploiting its long-term and periodic behaviour.

The resulting model combines the estimates delivered by the static and dynamic models in the following way:

$$f(s,t) = \frac{\lambda}{r_d(s,t)+\lambda} f_s(s,t) + \frac{r_d(s,t)}{r_d(s,t)+\lambda} f_d(s,t), \tag{1}$$

where $r_d(s,t) \geq 0$ is a reliability of the dynamic model f_d for a given segment s and a given time point t, and $\lambda \geq 0$ is a mixing parameter (tuned experimentally). The reliability defines our trust in the dynamic model. If there are only few or no recent observations, then r_d should be set to a value close to zero or to zero, respectively.

In the following, we use simple static and simple dynamic models to illustrate the capability of the combined model to accurately estimate travel times in the traffic network. Despite their simplicity, these two models, when combined, are powerful enough to be used in practical situations.

4.2 The Static and Dynamic Components

The simplest static model, referred to as the global mean, is based on global averaging:

$$f_s(s,t) = l(s) \times \frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} l(s_i)}, \tag{2}$$

where $l(s)$ is the length of the segment s.

The significant improvement of this model can be obtained by using the segment mean model, which averages the travel times on each road segment separately:

$$f_s(s,t) = \frac{\sum_{s_i=s} y_i}{\sum_{s_i=s} 1}. \tag{3}$$

The particular dynamic model f_d is constructed as a time series for each road segment. Prediction $f_d(s,t)$ for a given segment s and a given time point t is computed using previous observations y_{st_i}, $t_i < t$, from segment s. Training data are then represented for each segment $s \in \{1, \dots S\}$ in the form $(y_{st_1}, y_{st_2}, \dots, y_{st_{N_s}})$, where N_s is the number of observations for segment s. These observations are aggregated by being averaged over a given time interval T (tuned experimentally):

$$f_d(s,t) = \frac{\sum_{t-t_i<T} y_{st_i}}{\sum_{t-t_i<T} 1}. \tag{4}$$

The reliability parameter $r_d(s,t)$ of this model is set to the number of observations from T, i.e., to $(\sum_{t-t_i<T} 1)$. Thus, we can reformulate the final, combined model (1) to:

$$f(s,t) = \frac{\lambda f_s(s,t) + \sum_{t-t_i<T} y_{st_i}}{\lambda + \sum_{t-t_i<T} 1}, \tag{5}$$

which produces as its output a weighted average over the static model and the most recent observations.

4.3 Experimental Study

In the experiments, we use floating car data that cover the area of Poznań with broad surroundings. The area can be defined as a rectangular envelope with side lengths of above 60 km, centred at 52.3964°N 16.8421°E. In the time domain, the observations span three weeks of 2011: from September 12th till October 2nd, collected between 5:00 a.m. and the midnight (i.e. excluding night hours). The entire data set contains about 3.8 million observations. It should be stressed, however, that the observations are sparse and not evenly distributed in time and space.

We split the data into two parts: the training set and the test set. The training set covers the observations collected during the first two weeks, i.e. from September 12th till September 25th, and is used to construct the static model and to tune the λ parameter. The test set covers observations collected during the last week, i.e. from September 26th till October 2nd, and is used to test the overall performance of the models.

We use in total three methods for travel time estimation: the global mean (GM), the segment mean (SM), and the combination of the segment mean with the dynamic model (CM). We take the observations from the last 5, 15, 30, 60, 120 minutes for building the dynamic model and optimize λ in range $[0.0, 5.0]$ with step 0.5.

Table 2 Results of the three models on test set. Mean absolute (MAE) and root mean squared error (RMSE) are reported.

Model	MAE [min]	MAE [%]	RMSE [min]	RMSE [%]
GM	0.1818	100.0	0.5464	100.00
SM	0.1322	72.74	0.4710	86.20
CM, λ=0.0, $T = 5$ min	0.1322	72.74	0.4710	86.20
CM, λ=2.0, $T = 15$ min	0.1287	70.80	0.4567	83.58
CM, λ=2.0, $T = 30$ min	0.1260	69.33	0.4430	81.07
CM, λ=2.5, $T = 60$ min	0.1247	68.61	0.4357	79.73
CM, λ=4.0, $T = 120$ min	0.1255	69.05	0.4344	79.50

The results of the experiment are shown in Table 2. The table reveals that the segment mean improves significantly over the global mean, and the dynamic model improves further over the segment mean. This is due to the adaptive nature of the dynamic model. Interestingly, the best results are obtained for the time interval T equal to 120 minutes.

5 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 papers 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 combined model in travel time estimation). In both cases the more ad-

vanced methods significantly outperformed the simple ones, achieving results that make these methods useful enough to be used to practical applications, despite their increased computational demands.

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

References

1. Billings, D., Yang, J.: Application of the ARIMA models to urban roadway travel time prediction — a case study. In: IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC'06, vol. 3 (2006)
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B **39**(1), 1–38 (1997)
3. Hilligoss, B., Rieh, S.Y.: Developing a unifying framework of credibility assessment: Construct, heuristics, and interaction in context. Information Processing and Management **44**, 1467–1484 (2008)
4. Kubiak, M.: 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 (2007)
5. Liu, H., van Lint, H., van Zuylen, H., Zhang, K.: Two distinct ways of using Kalman filters to predict urban arterial travel time. In: Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, pp. 845–850. IEEE (2006)
6. Premaratne, K., Nunez, R., Wickramarathne, T., Murthi, M., Pravia, M., Kuebler, S., Scheutz, M.: Credibility assessment and inference for fusion of hard and soft information. In: Proceedings of AHFE (2012)
7. Rice, J., Van Zwet, E.: A simple and effective method for predicting travel times on freeways. Intelligent Transportation Systems, IEEE Transactions on **5**(3), 200–207 (2004)
8. Tseng, S., Fogg, B.J.: Credibility and computing technology. Communications of the ACM **42**(5), 39–44 (1999)
9. Van Lint, J., Hoogendoorn, S., Van Zuylen, H.: Accurate freeway travel time prediction with state-space neural networks under missing data. Transportation Research Part C **13**(5–6), 347–369 (2005)
10. Wan, K., Kornhauser, A.: Turn-by-turn routing decision based on copula travel time estimation with observable floating-car data. In: Transportation Research Board 89th Annual Meeting, 10-2723 (2010)
11. Zhu, T., Kong, X., Lv, W., Zhang, Y., Du, B.: Travel time prediction for float car system based on time series. In: Advanced Communication Technology (ICACT), 2010 The 12th International Conference on, vol. 2, pp. 1503–1508 (2010)