Location based learning of user behavior for proactive recommender systems in car comfort functions

Thomas Stone, Olga Birth, André Gensler, Andreas Huber, Martin Jänicke, Bernhard Sick

Universität Kassel
Intelligent Embedded Systems
{gensler,jaenicke,bsick}@uni-kassel.de

BMW Group
Forschung und Technik
Olga.Birth@bmw.de

BMW Group
Intelligente Komfortfunktionen
{Andreas.AH.Huber,Thomas.Stone}@bmw.de

Abstract:
In-car comfort functions decrease driving stress and therefore increase safety. Drivers typically do not use all available comfort functions optimally, if at all, in every situation they would offer a substantial increase in comfort. Automating such functions as proactive recommender systems would exploit the full potential for decreasing driver stress. Because comfort functions are highly dependent on the driver's habits, learning the individual user behavior is necessary. We propose a probabilistic method for modelling and predicting location dependent user behavior of comfort function activations. The model applies second-order uncertainty to evaluate the certainty about inferred parameter values and it deals with novelty and decaying observations explicitly. The results of this study show that the use of probabilistic models for learning location based user behavior in car comfort functions is a promising technique and gives reason to further investigate this area of studies.

1 Introduction

The primary research efforts in the area of vehicular automation is concentrated on context-aware ADAS. For automotive automation, context-awareness is an important issue, because many complex use cases become more dynamic and ambigous with a higher level of automation. Comfort functions also have potential for automation. Automating these functions eliminates or simplifies the user interaction, decreasing driver stress. As for now, most comfort functions' system parameters are set to fit the average user. The parameters are determined by user studies and expert knowledge, incorporating little to no online, user-specific contextual information. The driver's information, like calendar data, frequent routes, the vehicle's current location or the individual pattern of handling assistance and comfort systems, can improve the precision of automation and enable personalized features, therefore increasing the acceptance of such systems.

For the development of future automated comfort functions, learning the user's individual behavior is a vital part and makes personalized, proactive recommendations possible. As an example, the class of climatic comfort functions is used dependent on climatic conditions, personal preferences and environmental circumstances. For non-climatic functions, the vehicle's current geopraphic or road type based location may have a strong effect on the probability of the activation by the user, e.g., parking assistance systems may only be used close to free parking spaces. Systems which automate functions, depending on such locations of interest, are called location based learning systems. We will focus on learning locations of interest and dependent user function activations. Further on, using this knowledge for proactive recommendations will be called location based automation and recommendation and is the target application of the proposed algorithms.

There are several major requirements that any procative vehicular recommentation system must satisfy. The model must be embedded, soft real-time capable, operating economically with memory and processing time consumption. Also, a mechanism for self-reflection is necessary to detect when the certainty about inferred parameters is high enough to make proactive recommendations. A very important property is the concept of online and lifelong learning, which means that a system can cope with novel situations as well as already observed ones fast (a precise definition is given in [FH06]). To give the user the possibility to control the system directly up to a certain degree, implicit and explicit feedback must be taken into account by the system. Explicit feedback is a user action (e.g., activating an automated function or counteracting to an automation), while implicit feedback is the lack of user action at a certain location which can also mean to accept the actions of such a system. At last, the user must have an understanding about the system's mispredictions. In a highly subjective field such as comfort functions, it is barely possible to construct a never-failing prediction system.

In the following proposal, we will describe a system that will meet all of these goals, whether implicitly or through explicit modeling.

2 Related work

There has been a lot of research in the area of recommendation systems for increasing comfort in cars. Such systems include context-aware and proactive recommendations, such as optimal gas refueling recommendations [WBE09] or models for recognizing driver fatigue for trip pause suggestions [Yan07].

The systems presented in [GKFC11], [KGT$^+$08] and [APRR07] incorporate contextual information. Some of the information is retrieved through the internet, while others, e.g., the user's preferences, must be specified explicitly by the user.

Complex contextual models for calculating recommendations are presented in [BNWP11], [WY10] and [Yan07]. The incorporated model in [BNWP11] is based on Multi-Criteria Decision Making and Fuzzy Logic, yielding complex algorithms for a pseudo-probabilistic system. These systems do not have a self-reflection mechanism like second-order uncertainty.

There are systems that incorporate an explicit user model. [FPFA11] and [FMA12] define three main information sources for combining these into a recommendation, i.e., input information, context information and profile information. [WE07] defines a contextual recommendation system as a function $U \times I \times C \rightarrow R$, where U is the user and his information, I is a possible recommendation item, C is the contextual information and R is the rating for the recommendation. The recommendation system then chooses the best rated items for the recommendation. [WY10] and [Yan07] use Dynamic Bayesian Networks to model different states of a driver, e.g., the driver fatigue or driver attention.

In all these systems, important parameters are mostly set through expert knowledge, i.e., surveys of experts, and field studies. These systems do not adapt their parameters online to the user's habits, though such techniques exist in machine learning. Some of the presented algorithms use probabilistic, mostly pseudo-probabilistic, methods to overcome the problem of uncertain or missing knowledge. Also, some algorithms do not use any learning algorithms at all. Altogether, most of these systems use simple techniques for inference and data representation, that do not implement a self-reflection mechanism, which is a major issue in automotive recommender systems. This work introduces probabilistic algorithms with self-reflection capabilities and the ability to adapt parameters for an explicit user modelling. Also, our proposed algorithm does not need any prior information from the user and does not require the driver to interact more with the system than activating or deactivating the automated function.

3 Location based Learning

A location based learning system must implement several key concepts, such as observation of the user's actions, user feedback evaluation and the adaption of system parameters for user specific system behavior. A "learning system" in general can be defined as one that makes decisions upon previously gathered knowledge inferred from observations of the user's interaction.When having made enough observations, the system can start working autonomously or at least suggest actions to the user based on his past behavior.

In a vehicular environment, messages from a car data bus, e.g., CAN, serve as observations B. These messages may contain information about sensors, actuators, media data or other information distributed across different electronic control units. Beside location information, these observations also include activations A of functions triggered by the user that we want to automate. We use a probablistic approach to model the likeliness of activations in the system. The overall probability of an activation at an abstract location O (which can be translated to a more general definition of locations such as "at a certain intersection", "at home", etc.) is then built with

$$p(A|B) = \sum_O p(A,O|B), \tag{1}$$

$$= \sum_O p(A|B,O) \cdot p(O|B). \tag{2}$$

Depending on the value $p(A|B)$, the result can be interpreted as the certainty of the system for an activation A given an observation B. To specify our problem, we make the following assumptions:

1. The probability of an activation A given observation B and location O corresponds to the probability of an activation A given a location O, i. e., the information of an observation does not yield any further information if the location is known.

$$p(A|B,O) = p(A|O) \tag{3}$$

2. If an observation B is very likely at location O_j, all other locations are very unlikely.

$$\forall O_i : p(O_i|B) \approx 0, i \neq j \tag{4}$$

$$p(O_j|B) \approx 1 \tag{5}$$

Taking into account these assumptions, the overall probability for a location O_j can be formulated as the joint probability

$$p(A|B) = p(A|O_j) \cdot p(O_j|B). \tag{6}$$

The following sections describe the elements of this equation in more detail. The term $p(O|B)$, the *Location Distribution*, is described in Section 3.2. The term $p(A|O)$, the *Activation Distribution*, is detailed in Section 3.3. The adaption of these distributions is performed with user *feedback* (see Section 3.1) and is described first as the two models are based on some of the feedback assumptions.

3.1 Feedback

In general, a learning system has to combine two fundamentally different goals:

1. Stability: The system has to reliably recognize the user's wish and increase the security of the recognition over time with additional knowledge.

2. Plasticity: The system has to be able to react reasonably fast to a change in the user's usage pattern.

Consequently, a mechanism which increases its certainty over time (respectively with more observations) has to be used. Furthermore, it has to be able to adapt to changes in the users behavior, even after the system got sure with respect to its stability.

Feedback is the main mechanism to enable learning in our system and it applies to probablistic zones (each modeled with a distribution $p(A|B)$) at a location O which we will call *Activation Zones (AZ)*. The different forms of possible feedback depend on the (re-)action of the user and the current system state. Table 1 gives insight on the possible forms of interaction.

Case Nr.	AZ exists?	System Activation	Driver-(re-)action	Remarks	Adaption $p(A\|O)$	Adaption $p(O\|B)$
1	✗	✗	$\circ$	no learning request	$\circ$	✗
2	✗	✗	$+$	automation request (new AZ)	$++$	✓✓
3	✓	✗	$\circ$	no automation desired	$-$	✗
4	✓	✗	$+$	automation request	$++$	✓
5	✓	✓	$-$	counteraction	$--$	✗
6	✓	✓	$\circ$	acceptance	$+$	✗

Table 1: Feedback matrix: $\circ$ represents no action, $+$ represents positive (re)action, $-$ represents negative (re)action, double symbols represent stronger reactions and represent a feedback strength hierarchy.

If no information about the location O is present in the system, the corresponding cases of the feedback matrix are the cases 1 and 2. The default value of the system is case 1 (no activation, no known AZ), whereas case 2 represents a first activation of the observed activation A, which should lead to the creation of a new AZ to be observed. If the system enters an already observed AZ, cases $3-6$ are evaluated. The value of the activation distribution $p(A|O)$ (see Section 3.3 for details) decides whether the autonomous system activation is triggered or not (✓/ ✗). So, if the system is in case 3 (Activation: ✗, no driver action), the resulting activation probability of $p(A|B)$ will be lowered while the position distribution is not adapted. The same scheme applies to the cases $4-6$.

3.2 Location Distribution

The location distribution describes the probability of the car to be at a certain location O when making an observation B, i. e., the probability distribution $p(O|B)$. In the context of location based learning, locations where an activation A of a function to be automated is observed, are relevant. In general, the location O does not only have to incorporate geographic coordinates, but can be extended by other dimensions in order to define the location more precisely. O therefore can be seen as a state variable of the system. For the sake of explanation, we will stick to just the geographic coordinates here. Observations always occur at a singular position, by itself they do not have any spatial size. Still, they have to tolerate spatial neighborhood in order to be incorporated into a model. To model abstract locations O, spatial neighborhood is considered in an Activation Zone (AZ) by evaluating the distribution $p(O|B)$.

The distribution $p(O|B)$ depends on the distance between the observed point and the AZ. The further away the observed point from the AZ, the lower is the probability to be inside this zone. Taking into account the above assumptions Eq. 4 and Eq. 5, it makes sense to model each AZ independently. An approximation of the location distribution can be made by using a normal distribution in the form

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}. \tag{7}$$

In Eq. 7, $\mathbf{x}$ is the coordinate value of the current observation B, μ is the center of the AZ and Σ the covariance matrix describing the spatial form and size of the distribution. Here, $\mathbf{x}$ and μ are assumed to be two-dimensional observations (latitude, longitude), consequently the covariance matrix happens to be a symmetrical 2x2 matrix. For the sake of parameter reduction we simplify $\boldsymbol{\Sigma}$ to be an isotropic covariance matrix with only one degree of freedom in the form

$$\boldsymbol{\Sigma}_{iso} = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}. \tag{8}$$

The distance between the observation and the center of the AZ can then be determined via the Mahalanobis distance $\Delta(\mathbf{x})$ which is determined by

$$\Delta(\mathbf{x}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \tag{9}$$

$\Delta(\mathbf{x})$ can be compared with a threshold ρ in order to determine the belonging of an observation $\mathbf{x}$ to the AZ. An advantage of the normal distribution is that its parameters μ and Σ can be adapted using update formulas, so we do not have to store each observation separately.

The parameter ρ can be chosen by evaluating the χ^2 distribution as the Mahalanobis distance $\Delta(\mathbf{x})$ of a D-dimensional random variable $\mathbf{x}$ ($\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathbf{x} \in \mathbb{R}^D$) is χ^2_k distributed. k hereby represents the degrees of freedom for the distribution, so in our case $k = D = 2$ when considering 2D coordinates. We can then set ρ to a value which guarantees that a certain percentage of the values for $\mathbf{x}$ lies within the AZ. ρ therefore acts as an upper limit for the Mahalanobis distance.

For our location model we introduce an iterative location refinement which improves its accuracy with an increasing number of observations. If an activation of an observed variable occures and is not assigned to an already existing AZ, a new zone is created (see case 2 of Table 1). The first activation of this AZ (belonging to an abstract location O) does not necessarily have to be at the ideal zone center. Figure 1 (a), (b) visualizes the zone creation. The size of the zone (parameter $\boldsymbol{\Sigma}$) has to be initialized according to the requirements of the application.

If the system performs an automation request (case 4 of Table 1), the parameters of the location will be adjusted. As we can see from the table, only in case 4 of the feedback matrix an adaption of the location is performed. This is done by means of the current

(a) Initial situation

(b) Initialized Activation Zone

(c) Adaption Phase (I)

(d) Adaption Phase (II)

Figure 1: Creation of a new AZ by a single activation of the system. The real zone center (abstract location O) is marked by the black cross, the initialized zone center and size by the red cross and circle. Note that the new AZ (initialized using observation B) is not located at the location of O. Over time, the location center ought to increase its accuracy with the true location O in both center and size of the AZ.

observation $\mathbf{x}$ via a formula incorporating the the previous (two-dimensional) center $\mu_{\mathbf{i}}$ to calculate the new center $\mu_{\mathbf{i+1}}$. When applying the formula, the center of the AZ is shifted with each iteration.

Likewise, the values a of the covariance matrix $\mathbf{\Sigma}$ are assumed to be $a = \sigma_i^2$, the updated parameter σ_{i+1}^2 is then calculated via an update formula taking into account the previous variance σ_i^2 and the deviation of the current observation $\mathbf{x}$ from the center $\mu_{\mathbf{i}}$. Note that in both update formulas, only the current values of μ and σ^2 have to be stored, not every single observation of $\mathbf{x}$. It should be remarked that a limit for the minimum size of the AZ can be introduced to avoid overfitting to the data. An example of the iterative adaption of an AZ is shown in Figure 1 (b), (c), (d).

3.3 Activation Distribution

Every Activation Zone (AZ) has the associated position distribution $p(O|B)$ (see Section 3.2) and the activation distribution $p(A|O)$, which is described here. The type of activation distribution chosen is the Bernoulli distribution, which describes a random experiment that only has the outcomes *success* or *no success*. These outcomes can be translated to activation or no activation of the observed function when passing through an AZ. The probability distribution should not only return a probability p in the range $[0, 1]$ for the activation probability, but also be aware of the certainty the distribution has regarding the estimation of p. To additionally take these requirements into account, we use a distribution which is conjugated to the Bernoulli Distribution, the Beta Distribution [Bis06].

The Beta Distribution has two input parameters a and b, which again represent activation respectively no activation. The idea of this second-order technique is to treat the probabilistic parameter p of the Bernoulli distribution as a random variable itself and determine its mean $\mathbb{E}[\mathrm{p}]$ with

$$\mathbb{E}[\mathrm{p}] = \frac{a}{a+b}. \tag{10}$$

Furthermore, the variance of the Beta Distribution is defined by

$$\mathrm{Var}[\mathrm{p}] = \frac{ab}{(a+b+1)(a+b)^2}. \tag{11}$$

$\mathrm{Var}[\mathrm{p}]$ can be used as a measure for the certainty of the expected mean $\mathbb{E}[\mathrm{p}]$.

To recall, learning algorithms should take into account the goals stability and plasticity (Section 3.1). By just summing up (multiples of) a and b depending on the users action when passing through an AZ, we can guarantee stability, but the algorithm falls short on plasticity, i.e., the algorithm cannot react fast on sudden changes in the user's behavior.

To also incorporate plasticity, we introduce an adapted form of feedback handling which weights each observation $P(l)$ (where $l = 1$ is the most current observation) according to their order of occurrence (e.g., using an exponentially decreasing weighting function) with the weighting term $G(l)$ and sums them up to get the respective a and b. Hence, old values will get obsolete, consequently only a fixed number of N observations has to be stored in memory.

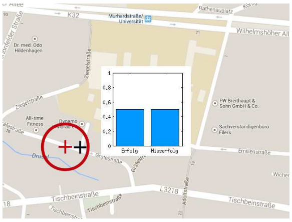
4 Case study

In a case study, the implemented algorithms were evaluated for a comfort function that is typically activated at specific geographic locations. The automated comfort function is a high-class optional equipment mostly sold in packages. For the feature selection, parameter determination and algorithm evaluation, test data had to be recorded in real-world trips. About ten BMW owners, regularly using the selected comfort function, participated in the data recordings, over a hundred trips from every participant were recorded on average. The car owners were instructed about the general goal of the recommendation system and to use the automated comfort function as naturally as possible. Some subjects not having any technical background, ensured results close to a real-world application of the system. For reasons of data privacy, only the necessary data was filtered on the vehicle bus and then recorded with the designated logging hardware. The data recorded contained the activations and deactivations of the comfort function, filtered and mapped GPS data, date and time information and possibly function related environmental data.

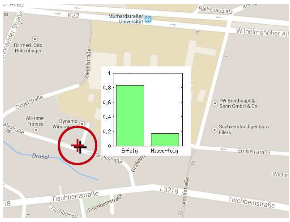
Step	$\mathbb{E}$[p]	Var[p]	Case of Table 1 in buffer $P(l)$, $l = 1, ..., N$									
			1			.	.	.	.			N
1	0.5	0.0227	2	2								
2	0.75	0.0089	4	2	2							
3	0.83	0.0045	4	4	2	2						
4	0.90	0.0018	4	4	4	2	2					
5	0.91	0.0015	6	4	4	4	2	2				
6	0.92	0.0013	6	6	4	4	4	2	2			
7	0.79	0.0024	5	6	6	4	4	4	2	2		
8	0.71	0.0026	3	5	6	6	4	4	4	2	2	
9	0.65	0.0027	3	3	5	6	6	4	4	4	2	2
10	0.64	0.0026	3	3	3	5	6	6	4	4	4	2
11	0.57	0.0028	3	3	3	3	5	6	6	4	4	4
12	0.47	0.0029	3	3	3	3	3	5	6	6	4	4

Table 2: The contents of an observations buffer of an AZ with 10 elements and 12 updates. The buffer is a FIFO buffer with new observations added on the left side.

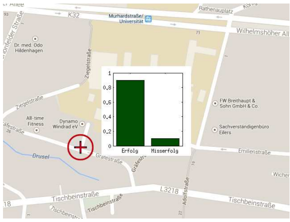
The following case study illustrates how the proposed system updates the activation distribution $p(A|O)$. For the sake of simplicity, we assume our weighting term $G(l)$ to be equally weighted, i.e., $G = [1, ..., 1]$, for the values of our feedback matrix cases we assume values following the feedback strength hierarchy (see Table 1). The example given in Table 2 and Figure 2 both showing the same experiment describes how the system updates the parameters a and b. At initialization, both parameters are set equally, i.e., $\mathbb{E}[\text{p}] = 0.5$, which is realized by adding two different forms of equally strong feedback to the buffer $P(l)$ (see Table 2 (1), Figure 2 (a)). If an activation request (case 4 of Table 1) is observed, the value of a rises while b stays constant (Table 2 (2)-(4), Figures 2 (b), (c)). At the same time, the position distribution $p(O|B)$ is also refined (see Figures 2 (b), (c)). It is worth noting, that the variance, which is given to express parameter uncertainty, decreases the more observations are made by the system. In our case study, we request the activation threshold for $\mathbb{E}$[p] to be greater or equal a threshold $\tau = 0.9$, which is achieved in Table 2 (4). The system now automatically activates the comfort function when the driver passes the AZ. If the user accepts the systems behavior, i.e., he does not counteract to it (case 6 of Table 1), the system continues to improve its mean $\mathbb{E}[p]$ (Table 2 (5), (6), Figure 2 (d)). In our case study, the user now changes his usage pattern, counteracting the automatic system activation (case 5 of Table 1). This leads to a decrease of the estimated mean $\mathbb{E}[p]$ below the threshold τ. (Table 2 (7), Figure 2 (e)). When the behavior of the driver changes, Var[p] increases slightly, expressing increased uncertainty about the estimated mean $\mathbb{E}[p]$. The system will now stop to activate the automated function, if the user continues to not activate the system at the specified location (Table 2 (8) - (12)), $\mathbb{E}[p]$ will continue decreasing (case 3 of Table 1). Over time, the estimated mean $\mathbb{E}[p]$ may get as low as a predefined obsolescence threshold so that the AZ will get deleted if not relevant (Figure 2 (f)).



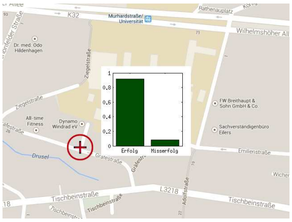
(a) AZ initialization, $\mathbb{E}[p] = 0{,}5$



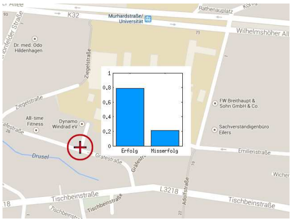
(b) after two automation requests, $\mathbb{E}[p] \approx 0{,}83$, $\mathbb{E}[p] < \tau$



(c) AZ after three automation requests (system now active), $\mathbb{E}[p] = 0{,}90$, $\mathbb{E}[p] \geq \tau$



(d) AZ after two acceptances, $\mathbb{E}[p] \approx 0{,}92$



(e) AZ after counter action, $\mathbb{E}[p] = 0{,}79$, $\mathbb{E}[p] < \tau$

(f) AZ deleted (under obsolescence threshold)

Figure 2: Example of activation updates of an activation zone (AZ). In the histogram of (c), the activation threshold τ is exceeded. In figures (a) – (c) the adaption of the location density distribution is shown.

The processing time is reasonably small, as most of the time only the position distribution $p(O|B)$ has to be calculated (which mainly consists of the computation of the Mahalanobis distance $\Delta(x)$). The computing time scales linearly with the number of AZ. The memory footprint is small enough for embedded use, as only the buffer of activations $P(l)$, the spatial coordinates μ, the AZ size σ and the number of position refinements i have to be stored.

5 Conclusion

The introduced probabilistic system predicts a user activation request for a specified comfort function depending on previously observed activations at points of interest. The probability distribution of the activation and the locations of interest are modelled explicitly. With the use of second-order uncertainty, it implements the necessary self-reflecting mechanism useful for automotive needs. Also, there is no need for the user to explicitly give any information for initialization to the fully automated system. Therefore, this system is appropriate for decision making in a proactive recommender system and can be adapted to any kind of comfort function.

The given evaluations show that the system works efficiently, is numerically stable and has enough plasticity to work as an embedded, online and life-long learning algorithm. It is also possible to change the model's plasticity and stability further through adjustments to the employed system parameters. Before field operation, the appropriate system parameters have to be discovered through a long-term study with a high number of subjects.

6 Outlook

In future work we will enhance the presented model's targeted set of functions. A focus of this enhancement will be the correlation and dependency of function activations, e.g., the usage of certain heating functions may correlate to a certain temperature or the usage of other comfort functions. As a consequence, the proposed approach for mobile, context based systems in [SBG98] will be applied.

Until now, the model distinguishes only between binary activation states. In future versions of the presented model, specific levels of activation, that in a sense have an ordering, will be handled to automate even more comfort functions. We also aim to extend our model to gradual output levels.

Also, we will enhance the model's observation B with more features. Part of the features incorporated will be the day time and the drive purpose of frequent trips, e.g., the daily trip to work. This should give the model information about the user's control behavior, e.g., the purpose of a trip or even environmental conditions. In future work, more general location models will be examined.

As for now, for complex recommendation models, some system parameters, e.g., decision thresholds, must be set by experts or through knowledge gained from field studies. This sometimes is a vague and expensive process. So the process of choosing the right parameters must be simplified or partially transferred into the field. In general, a wide field of research possibilities are models that change the system parameters online to better accommodate to individual users behavior. One example would be a collaborative filtering approach for online parameter refinement in proactive, automotive recommendation systems.

References

[APRR07] M. Ablaßmeier, T. Poitschke, S. Reifinger, and G. Rigoll. Context-aware information agents for the automotive domain using Bayesian networks. In *Proceedings of the Symposium on Human Interface: Part I*, pages 561–570, 2007.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Secaucus, USA, 2006.

[BNWP11] R. Bader, E. Neufeld, W. Woerndl, and V. Prinz. Context-Aware POI Recommendations in an Automotive Scenario using Multi-Criteria Decision Making Methods. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation, Palo Alto, USA*, pages 23–30, 2011.

[FH06] S. Furao and O. Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106, 2006.

[FMA12] J. Ferreira, V. Monteiro, and J. Afonso. Electric vehicle route recommender system. In *International Conference on Smart Grids and IT Systems, Chengdu, CN*, volume 1, pages 1–10, 2012.

[FPFA11] J. Ferreira, P. Pereira, P. Filipe, and J. Afonso. Recommender system for drivers of electric vehicles. In *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT'11), Lisbon ,PT*, 2011.

[GKFC11] O. Gusikhin, E. Klampfl, D. Filev, and Y. Chen. Emotive Driver Advisor System (EDAS). In *Informatics in Control, Automation and Robotics*, volume 89 of *Lecture Notes in Electrical Engineering (LNEE)*, pages 21–36. Springer Berlin Heidelberg, 2011.

[KGT+08] E. Klampfi, O. Gusikhin, K. Theisen, Y. Liu, and Giuli T.J. Intelligent Refueling Advisory System. In *Proceedings of the second International Workshop in Intelligent Vehicle, Madeira, POR*, volume 2, pages 60–72, 2008.

[SBG98] A. Schmidt, M. Beigl, and H. Gellersen. There is more to Context than Location. *Computers and Graphics*, 23:893–901, 1998.

[WBE09] W. Woerndl, M. Brocco, and R. Eigner. Context-Aware Recommender Systems in Mobile Scenarios. *International Journal of Information Technology and Web Engineering*, 4(1):67–85, 2009.

[WE07] W. Woerndl and R. Eigner. Context-Aware, Collaborative Applications for Inter-Networked Cars. In *16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'07), Paris, FR*, volume 16, pages 180–185, 2007.

[WY10] Q. Wu and W. Yu. A Driver Abnormality Recondition Model Based On Dynamic Bayesian Network for Ubiquitous Computing. In *Third International Conference on Advanced Computer Theory and Engineering (ICACTE'10), Chengdu, CN*, volume 3, pages 320–324, 2010.

[Yan07] J. H. Yang. *Analysis and Detection of Driver Fatigue Caused by Sleep Deprivation*. PhD thesis, Massachusetts Institute of Technology, 2007.