# Car Drive Classification and Context Recognition for Personalized Entertainment Preference Learning

Thomas Christian Stone
BMW Group
Munich, Germany
Email: thomas.stone@bmw.de

Stefan Haas
SAP SE
Munich, Germany
Email: stefan.haas02@sap.com

Sarah Breitenstein
BMW Group
Munich, Germany
Email: sarah.breitenstein@bmw.de

Kevin Wiesner
Institute for Informatics
LMU Munich
Munich, Germany
Email: kevin.wiesner@ifi.lmu.de

Bernhard Sick
Intelligent Embedded Systems
University of Kassel
Kassel, Germany
Email: bsick@uni-kassel.de

***Abstract*—The automotive domain, with its increasing number of comfort and infotainment functions, offers a field of opportunities for pervasive and context-aware personalization. This can range from simple recommendations up to fully automated systems, depending on the information available. In this respect, frequent trips of individual drivers provide promising and interesting features, on the basis of which, usage patterns may possibly be learned and automated. This automation of functions could increase safety as well as comfort, as the driver can concentrate more on the experience of driving instead of repeatedly and manually adjusting comfort- or entertainment-related systems. To identify frequent driving contexts in a set of recorded signal in a vehicle, e.g., GPS tracks, this paper presents two different clustering algorithms: First, a hierarchical *Drive-Clustering*, which combines drives based on their number of common GPS points. Second, a *Start-Stop-Clustering*, which combines trips with the same start- and stop-cluster utilizing density based clustering. The *Start-Stop-Clustering* showed particularly good results, as it does not depend on the concrete routes taken to a stop position and it is still able to detect more trip clusters. To predict these drives, a Bayesian network is presented and evaluated, with logged trip data of 21 drivers. The Bayes Net uses context information, i.e., the time, weekday and the number of people in the car, to predict the most likely drive context with high accuracy. A new automated entertainment source selection algorithm demonstrates the usefulness of the retrieved information. The algorithm learns and predicts a driver's preferences for selected entertainment sources depending on recognized drive contexts.**

***Keywords–Context-aware Vehicle; Spatial Clustering; Drive Context Prediction; In-Car Infotainment; Automation***

## I. Introduction

Many different definitions for *context* exist, depending on the domain and conception. In [1], the frequent drives of a car owner considered *contextual information* useful for vehicle personalization.

In common literature, there seems to be a general notion for the meaning of the term *context*. However, up until now, there is no single definition accepted as the common standard. In [2], *context* is described as "…any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [3] claims to have found over a 150 definitions for *context*. Despite lacking a single, universally accepted definition, there is no argument about it's usefulness in certain applications. Context-awareness is considered an important building block in the development of intelligent systems as it is said to significantly improve the interaction between a user and a system. Knowledge about a specific context is normally gathered by sensor readings and their interpretation [4], [5]. In the course of this article, *context* will be considered any piece of information that can be aggregated in a vehicle and enables "intelligent behavior" of in-car systems. This includes not only information such as daytime, weekday, number of passengers, fuel level and frequent trip targets, but also the driver's control behavior in terms of the car's functions.

With its increasing number of comfort and infotainment functions, the automotive domain offers a unique field of opportunities for context-sensitive functions. In recent years, many different context-aware advanced driver assistance systems (ADAS) have been introduced. They depend on information provided by dedicated sensor systems, particularly in the areas of safety and comfort. The lane departure warning system (LDW), adaptive cruise control (ACC) and intelligent speed adaption (ISA) are well-known examples for context-aware ADAS.

Another interesting and promising context to advance vehicle personalization is the drive itself. Above all, the repeated drives of a person offer a lot of potential for finding consistent usage patterns. Subsequently, the found user behavior can be used for automating certain comfort functions. For example, if a driver usually checks received emails on the way to work or likes to listen to the news, the vehicle could adapt to these preferences by recognizing the drive context as a regularly drive to work and by automating the desired functions. This automation of functions could improve safety as well as comfort because the driver is no longer forced to adjust his personal settings by himself.

In the following, we will describe and evaluate different methods for the detection and prediction of repeated drives of individual drivers. To develop and evaluate our proposed methods, we had the possibility of utilizing recorded vehicle sensor data of 21 drivers collected over several months by a data logger. The collected data includes many different sensor signals exchanged between the different in-car electronic control units (ECU) over the Controller Area Network (CAN) bus, ranging from Global Positioning System (GPS) position to seat belt status.

The contributions of our article are two novel clustering methods for detecting repeated trips of individual drivers, a novel distance measure based on the Jaccard distance for comparing GPS tracks and a hybrid Bayesian network for predicting frequent drive contexts right away from the start of the trip based on contextual information, e.g., the time of the day and the number of passengers in the car. The frequent drives and the additional context information will be used to infer the intention of drives, e.g., ”drive to work” or ”drive to spare time activity”, what we consider as the *drive context*.

The article is structured as follows. Section II gives an overview on existing work in the fields of route prediction, route recognition, destination prediction, and place mining. Section III outlines two new spatial clustering methods for detecting the frequent drive contexts of a particular driver. In Section IV, we present a hybrid Bayesian network to predict the frequent drive contexts of an individual driver immediately from the start of the trip, or even during a trip. The results we obtained running the algorithms individually on the collected drive data of each driver are described in Section V. Additionally, in Section VI, we provide a case study for in-car infotainment automation based on the presented algorithms. The results prove our claim about the usefulness of the presented algorithm. We close our work in Section VII with a summary and an outlook on possible future work.

## II. Related work

Route recognition and prediction systems have been proposed in many different works [6], [7], [8], [9], [10]. In the majority of these publications, the general way to predict, and respectively recognize, the current route is based on the comparison of the current driving trajectory to previously recorded trajectories through suitable distance measures. Comparing GPS tracks can not be done with classic $L_p$ metrics due to their length related inequality, dimension and noise. As a result, more elastic similarity measures are necessary. Already proposed distance measures are, for instance, based on the longest common sub-sequence (LCSS) algorithm [6], [11], [12], the Hausdorff distance [7] or the Jaccard distance [13]. In [11], this simple instance based learning approach of comparing the current route to already recorded routes is further enhanced by the inclusion of contextual information, e.g., time of the day, to better differentiate overlapping routes.

Probabilistic approaches for route and destination prediction have been presented amongst others in [13], [14], [15] and [16]. The investigated prediction methods are frequently based on Bayesian techniques and include additional contextual information, such as the time of the day, the particular weekday or even background information about locations to infer the most likely route or destination [16]. By contrast, [15] uses an unspecified type of Markov model instead of a Bayesian approach to predict the next location of a user.

Identifying personally important places of users in recorded GPS data has, for example been investigated in [17], [18], [19], [15], [10] and [20]. Density based clustering hereby proved more efficient than classic partitioning algorithms like k-means [21], [22], [17], [18], as the final clusters only consist of dense regions in the data space. Regions of low object density are not included in the final clusters and are considered as noise.

Also, there has been work on using location based contextual information in proactive recommendation and automation. In [20], a probabilistic approach was presented for learning individual locations of interest. The learned locations were then used for recommendation and automation of a vehicular comfort function. The approach of learning an explicit user preference model proves helpful, especially for integration of user feedback and uncertainty quantification.

Our work differs from existing publications, as we focus on the personal, repeated drives of individual drivers and their prediction. This helps recognizing individual drive contexts. The drive contexts themselves denote regular drives, e.g., ”drive from home to work” or ”drive from work to home”. We consider the drive contexts as the basis for learning a driver's control behavior of certain functions. Therefore, the learned behavior is useful for recommendation and automation, which we will prove in a short-term study.

## III. Detecting frequent drives

The basis for drive context recognition will be the frequent drives of a driver. To detect frequent drive clusters of an individual driver, we present and evaluate two different spatial clustering methods explained in the following two subsections. *Drive-Clustering* is based on the Jaccard distance and compares whole trajectories using hierarchical clustering. In contrast, *Start-Stop-Clustering* focuses on more semantic similarity measurement of routes, based on the determination of frequent start and stop positions of a particular driver. The goal of both algorithms is to identify repeated patterns in the set of recorded GPS tracks in order to detect repeatedly occurring drive contexts, e.g., drives from home to work. In Section V, we compare the obtained results of both algorithms applied to our test data set.

### A. Drive-Clustering

An important factor in cluster analysis is a similarity measure to determine the distances between elements contained in the data, for the purpose of grouping similar elements together in clusters. In trajectory data the standard way for identifying patterns is to compare whole trajectories. In our case, the trajectory data of each drive is stored as a sequence of GPS points $S_i = \{p_{i,1}, p_{i,2}, ..., p_{i,n}\}$, with $p_{i,1}$ being the start point of the drive and $p_{i,n}$ being the end or stop point.

To compare two point sequences we use a dissimilarity measure based on the well known Jaccard distance, which measures dissimilarity between sample sets [23] (see equation (1)):

$$d(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}. \quad (1)$$

Our dissimilarity measure thereby calculates the intersection of the two GPS sequences $S_i$ and $S_j$ by counting the number of common points $NOCP(S_i, S_j)$ contained in both sequences starting from the shorter sequence (see equation (2)). This number of common points is then divided by the number of points contained in the shorter sequence $min(|S_i|, |S_j|)$. In order to obtain a dissimilarity measure the quotient is subtracted from 1, so that a result of 0 indicates maximum similarity and a value of 1 maximum dissimilarity.

$$d(S_i, S_j) = 1 - \frac{NOCP(S_i, S_j)}{min(|S_i|, |S_j|)}. \quad (2)$$

GPS points of two geometrically similar trajectories are unlikely to have exactly the same coordinates. Even if a drive can be done exactly the same way several times, the GPS sequences will not be equal because of the noisy nature of GPS measurements. Hence, it is necessary to define a threshold distance Θ, e.g., 50 meters, to decide whether two GPS points from two sequences can be considered as ”equal”. This is necessary to find the GPS points shared by both sequences, i.e., the *”common points”*. The threshold needs to be defined dependent on the logging frequency, assumed the GPS measurements were done periodically. In our case, the logging frequency is $f = 1Hz$. If we, for example, consider 135 km/h as the maximum vehicle speed, the maximum distance between two subsequent measured GPS points will be $(135 \cdot 1000)m/3600s = 37.5m$. In the evaluation we set the threshold to 50 meters, which is sufficient for driving speeds up to 180 km/h with a logging frequency of $f = 1Hz$.

The number of common points (NOCP) algorithm iterates over all points $p_{i,k} \in S_i$ included in the shorter sequence and tries to find at least one point in the other sequence $p_{j,l} \in S_j$ whose distance is less or equal than the defined threshold distance Θ. If the set of found points in range is not empty, the number of common points counter is increased. Consequently, the presented distance measure is more elastic than distance measures based on dynamic programming, such as the longest common sub-sequence (LCSS) or dynamic time warping (DTW), as it is able to match several elements of one sequence to just one element of the other sequence, without taking into account the sequence ordering. This behavior is important in our case to handle traffic jams and different driving speeds. The implementation of the number of common points (NOCP) function can be significantly sped up by storing the queried sequences' points in a *k-d tree* [24].

To calculate the distance between two-dimensional GPS points we use a simplification of the *haversine* formula [25] based on the Euclidean distance, which in contrast to the standard Euclidean distance allows metric parametrization of our algorithms ($\phi$ latitude, $\lambda$ longitude) (see equation (3)).

$$\begin{aligned} dist(\phi_1, \lambda_1, \phi_2, \lambda_2) = (((111.3 \cdot \cos(\frac{\phi_1 + \phi_2}{2}) \cdot \\ (\lambda_1 - \lambda_2)^2) + (111.3 \cdot (\phi_1 - \phi_2)^2))^{\frac{1}{2}} \cdot 1000. \end{aligned} \quad (3)$$

The *haversine* formula calculates the distance of two points on a sphere along their respective great-circle. While the original *haversine* formula is costly to calculate with all it's trigonometric functions, the given approximation is fast and precise for world GPS coordinates.

In order to avoid the problem of a much shorter sequence being contained in a longer sequence and to speed up the comparison, the number of common points in the two sequences is only calculated, when the first and last points of the two sequences are sufficiently similar. This means their respective distances do not exceed a predefined threshold, e.g., 250 meters ($p_{i,1} \sim p_{j,1}$ and $p_{i,n} \sim p_{j,m}$). Otherwise, the maximum dissimilarity value 1 is returned without any further calculation (see equation (4)).

$$d_{opt}(S_i, S_j) = \begin{cases} 1 - \frac{NOCP(S_i,S_j)}{min(S_i,S_j)}, & \text{if } p_{i,1} \sim p_{j,1} \\ & \text{and } p_{i,n} \sim p_{j,m} \\ 1, & \text{otherwise} \end{cases} . \quad (4)$$

To group similar routes in clusters, we use *single-linkage clustering*, an *agglomerative hierarchical clustering method*, starting from single GPS sequences. We use the function $d_{opt}$ for distance measurement. A merging threshold $\varepsilon$ decides whether two clusters are close or similar enough to be merged, e.g., $\varepsilon = 0.05$. The clustering stops when there are no clusters left for merging. The smaller the value $\varepsilon$ the more similar the trips contained in a cluster are, but in general, less clusters will be merged. This threshold will cut the *dendrogram* at a certain level and lead to the final drive clusters. The resulting clusters without enough observations can be considered *outliers* and will be deleted. To predefine the minimum cluster size we use another parameter $MinDrives$. As every point in our clusters represents a single drive, $MinDrives$ represents the parameter $MinPoints$ introduced in the density based clustering in [22]. This renaming was done for the purpose of convenience.

### B. Start-Stop-Clustering

Another way of determining frequent drives of a certain driver is based on his frequent start and stop positions of drives. The start and stop positions are the GPS locations, where the car is started and parked respectively. In contrast to the above presented trajectory clustering method, this method focuses on drives with the same start and stop positions, not on geometrically similar routes.

As the vehicle is typically not parked at the exact same coordinates, it is necessary to merge similar parking positions to *start-stop-clusters*. To obtain these frequent start and stop position clusters of a particular driver, we use a density based clustering, the DJ-Cluster algorithm presented in [17], which is a simplification of DBSCAN [22], [26]. Density based clustering has the advantage of explicitly eliminating outlier points compared to partitioning clustering, e.g., k-means [21], [26]. As we are only interested in dense regions included in the set of start and stop positions of an individual driver in order to identify frequent drive contexts, density based clustering is suitable for our task.

Consequently, the first step in Start-Stop-Clustering is to calculate dense regions of start and stop positions in the set of GPS sequences and to store the cluster IDs of every GPS sequences' start and stop points. Therefore, it is necessary to specify the two parameters $MinDrives$ and $\varepsilon$, representing the minimum cluster size and search radius respectively. Figure

1 shows an example of a dense point cluster found in the drive data of a particular driver with $\varepsilon = 100$m.

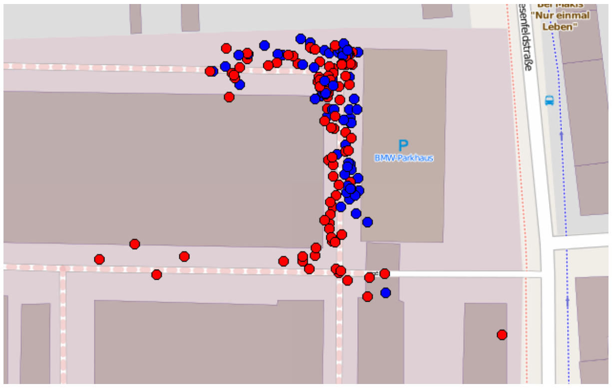


Figure 1. Visualization of the start (red) and stop points (blue) of a driver. All shown points are included in the same point cluster.

The equality of GPS sequences for Start-Stop-Clustering is determined by a binary function (see equation (5)).

$$d(S_i, S_j) = \begin{cases} 0, & \text{if } C_s(p_{i,1}) = C_s(p_{j,1}) \\ & \text{and } C_e(p_{i,n}) = C_e(p_{j,m}) \\ 1, & \text{otherwise} \end{cases} . \qquad (5)$$

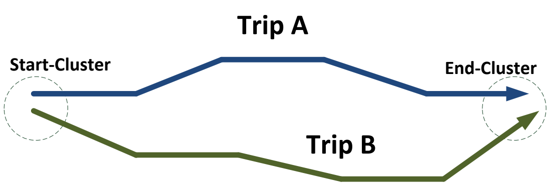



Figure 2. Illustration of a route-independent *Start-Stop-Cluster*.

Two GPS sequences $S_i$ and $S_j$ are considered as equal, when their corresponding start ($p_{i,1}, p_{j,1}$) and stop points ($p_{i,n}$, $p_{j,m}$) lie in the same start $C_s$, respectively end cluster $C_e$. Hence, the final frequent drive clusters are comprised of GPS sequences whose start and stop points lie in the same dense region or point cluster and therefore have the same cluster IDs. The clusters are direction-dependent just like those obtained with the above presented Drive-Clustering approach. However, the drives included in a *Start-Stop-Clustering* drive context cluster do not necessarily follow the same routes (see Figure 2). To predefine the minimum cluster size we also use the $MinDrives$ parameter.

## IV. Predicting Frequent Drive Contexts

The frequent drive clusters will be merged with additional information, forming a new context we will call *frequent drive context*. A *drive context* denotes a contextual description, or at least a semantic clustering of drives, depending on the intention of the drive, e.g., "drive to work" or "drive to gas station". *Frequent drive contexts* denote drives that happen to be periodic or frequent in a sense, e.g., daily drive to work and back home. As we will concentrate on frequent drives from now on, we will refer to them simply as *drive contexts*.

To predict frequent drive contexts that have been identified with one of the above presented methods, we propose a hybrid Bayesian network, incorporating more than just the clustered frequent drives as location based features. This is a basic requirement stated in [27], where contexts should not only consist of location based features. The structure of the network is shown in Figure 3.

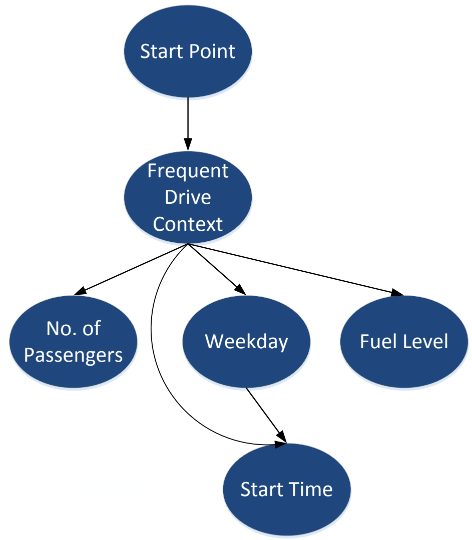



Figure 3. Topology of the hybrid Bayesian network for predicting the most likely frequent drive context.

Using the start point of the drive we are able to eliminate impossible contexts, e.g., a drive from work to home if the start point is home, which significantly reduces the possible contexts, prevents false positives and speeds up the implementation. The variable *Frequent Drive/Context* represents the *a priori* probability distribution over the set of identified drive contexts, already constrained by the current start point. The variables *Weekday*, *No. of Passengers* and *Fuel level* are conditionally independent of each other given the *class* variable *Frequent Drive Cluster*. The variables described so far all underlie a discrete probability distribution. The fuel level is discretized to "at least half full" and "not half full", in order to ease modeling. The number of passengers are discretized to $1, 2, 3, 4, 5$.

In contrast to the other probability variables, we model the variable *Start Time* as a continuous variable. By the edges between *Frequent Drive/Context*, *Day* and *Start Time* we receive a drive context dependent start time *probability density function* (PDF) for every single day. This enables a stronger differentiation between the drive contexts, as the start time probabilities for the different contexts are also dependent on the day.

To approximate the probability density function for the start times associated with a certain drive context we use *kernel density estimation* (KDE) (equation (6)) with a Gaussian kernel (equation (7)) and Scott's *rule of thumb* (equation (8)) for bandwidth selection $h$ [28]:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}). \qquad (6)$$

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \quad (7)$$

$$h_{scott} = n^{-1/(d+4)}. \quad (8)$$

$n$ is the number of observed starting times for approximating the underlying probability distribution. $K \in [0, 24] \rightarrow \mathbb{R}_0^+$ defines the kernel function used in the estimator, with $x \in [0, 24]$ being the daytime and $d$ the dimensionality of the problem.

By using *kernel density estimation*, we receive a continuous approximation of the probability density of starting points (see Figure 4).

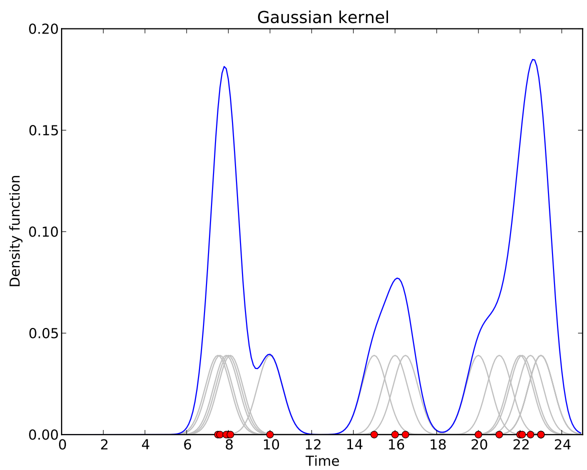



Figure 4. Example of a probability density function for the *Start Time* variable of a particular drive context.

When a drive context has not occurred before, at a certain day or time, the probability for the whole context will be zero. This kind of behavior is not always acceptable. A *Laplacian correction*, also called *Laplacian estimator*, is a common technique to solve this problem, as it adds observations to the dataset for unseen entities to prevent zero probabilities. We deliberately do not use *Laplacian correction*, thereby improving on false positives. In general, we assume a long-term observation phase for our proposed system, effectively eliminating this problem.

The probability for a certain context $C$, given the start point $s$, the weekday $d$, the time $t$, the number of persons in the car $p$ and the fuel level $f$, can then be calculated with the following formula:

$$P(C|s, d, t, p, f) \propto \\ P(C|s)P(d|C)P(t|d, C)P(p|C)P(f|C). \quad (9)$$

The context $C_i$ leading to the highest probability value $P(C_i|s, d, t, p, f)$ is then assumed to be the present context:

$$\arg\max_{C_i} \{P(C_i|s, d, t, p, f)\}. \quad (10)$$

The prediction in the presented form lacks the possibility to make online adaptions and corrections of the predictions. This may be necessary due to ambiguous information at the beginning of a trip, which prevents a good prediction of the current drive context. In the case where the starting points do not deliver enough information to predict the frequent drive cluster, the Bayes Net can be re-evaluated when the drive cluster is known during the trip. For instance, this can be achieved by constantly calculating the similarity of the current route to the frequent drive clusters. To that end, the similarity measure from Section III-A can be used. If the predicted frequent drive class of the current drive changes, the presented Bayes Net re-predicts the current drive context.

## V. Evaluation

To evaluate the described methods, we have access to a data set collected from 21 drivers over several months. The logger used for collecting the data, records all kinds of data bus traffic, also when the car is not moved, e.g., when the electronic key is pressed. To filter out this unwanted "noise", we only used recorded data for our evaluation where the vehicle was at least moved 1 kilometer (air-line distance). The number of filtered drives per driver ranged from 225 to 983 drives. This wide-spread distribution is related to the 3 to 8 months of recording duration and the individual use of the cars. The majority of the subjects ranged between 400 to 600 recorded drives.

### A. Drive clustering

Figures 5 and 6 show the results obtained applying Start-Stop-Clustering and Drive-Clustering to the data set. Figure 5 illustrates the average number of found clusters for different minimum cluster sizes ($MinDrives = \{3, 5, 10\}$). Figure 6 presents the average share of frequent drives of the total quantity of drives, i.e., "frequent" and "non-frequent" drives. As no ground truth could be gathered, the analysis must follow qualitative and quantitative reasoning.

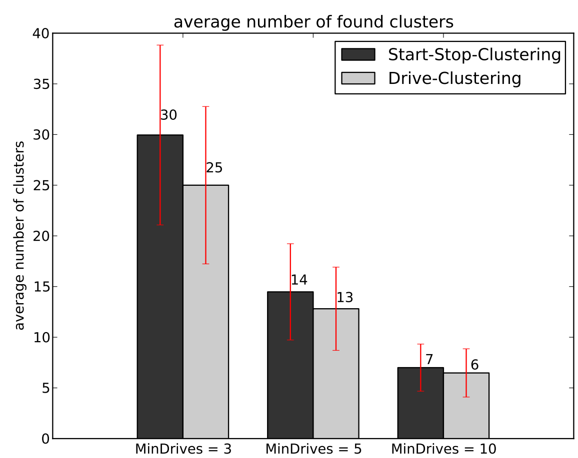



Figure 5. Average number of found clusters with Start-Stop- and Drive-Clustering dependent on the minimum number of drives contained in the clusters ($MinDrives$).

As one can see, Start-Stop-Clustering is on an average able to identify more clusters than Drive-Clustering (see Figure 5). However, with increasing minimum cluster size, the difference between the average number of found clusters by Start-Stop-Clustering and Drive-Clustering decreases. This leads to the

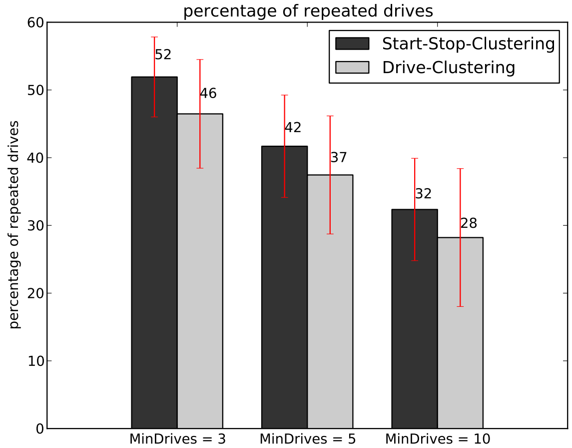



Figure 6. Percentage of repeated drives identified with Start-Stop- and Drive-Clustering dependent on the minimum number of drives contained in the clusters ($MinDrives$).

assumption that for frequent drives ($MinDrives = 10$), drivers usually have a preferred route that they normally take, whereas for less frequent drives ($MinDrives = 3$) they also take different routes to the same destination. Furthermore, Start-Stop-Clustering includes all route alternatives (see Figure 6), thus, assigning a larger fraction of the overall number of drives to a repeated drive cluster.

As we are rather interested in detecting frequent drive contexts than the frequent routes taken by a driver, Start-Stop-Clustering is more appropriate for our use case. Large clusters ($MinDrives \geq 10$) may provide promising and interesting contexts, on the basis of which usage patterns may possibly be learned and automated. The average fraction of trips, repeated at least 10 times by the participants during the survey, amounts to approximately 30% of the overall trips (see Figure 6).

To keep the set of frequent driving contexts up-to-date one could use a shifting time frame and only consider drives for the cluster calculation that for example occurred during the last 6 months. This would lead to a slow exclusion of no longer appearing driving contexts over time and also limit the amount of data used for the context identification.

### *B. Prediction*

To evaluate our proposed Bayesian inference system for predicting frequent drive contexts, we made use of cross-validation and focused on clusters identified by Start-Stop-Clustering with a cluster size larger than 10 drives. The cross-validation was done for every driver to even out the different recording times and to be able to differentiate between different types of drivers. The applied evaluation method was leave-one-out cross-validation to cope with the small data sets.

Figure 7 shows the overall prediction result for all drives, including also non-frequent drives, as well as the prediction result for solely frequent drives belonging to a cluster. The prediction result improves significantly, to almost 100% ($\sim$97%), when a prediction result is considered correct when lying within the top 3 predictions.

Evaluating the top 3 results shows the usability in recommendation systems. Presenting the user a recommendation for each of the $n$ most likely drive contexts is a common setup. In the case of recommendation based on learned user preferences, showing the most likely recommendation to the most likely contexts leads to a higher chance of user acceptance.

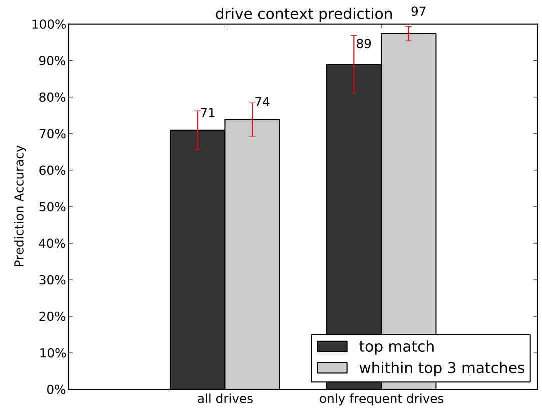



Figure 7. Prediction result for all drives and only frequent drive contexts ($MinDrives = 10$).

The differentiation between the different drive contexts is relatively accurate ($\sim$ 89% respectively $\sim$97% for top 3 matches). Moreover, in Figure 8 one can see that, when considering all drives, the main share in false predictions not lying within the top 3 matches is produced by false positives. A false positive is the classification of a starting drive as a "frequent drive", when it is not. A large fraction of false positives could be detected correctly ($\sim$60%), but as there might be highly frequent start and stop positions, e.g., home coordinates, with overlapping context information, e.g., time and weekday, some infrequent drives were predicted as belonging to a frequent drive context.

In the evaluation, we used a binary probability distribution for the day variable (workday, weekend) due to the relatively small minimal cluster size of 10 drives. It might be possible to achieve a better recognition of infrequent drives by assuming a discrete probability distribution for every day (Monday, Tuesday, Wednesday, etc.), which would also lead to time probabilities for every day for each drive context. However, this would only make sense with a higher minimal cluster size, in order to get representative probability distributions for every day.

Compared to the rate of false positives the rate of true negatives is extremely low and underlines the accuracy of our inference system related to the prediction of frequent drive contexts (see Figure 8). However, eliminating false positives is crucial in order to not annoy the driver with unwanted function automation and might only be solvable with little driver interaction. A solution could be to provide the driver with the top 3 most likely contexts. Then, the driver is able to choose the most appropriate one. If none is selected by the driver after a certain driving time, the system assumes that, in the current situation, no function automation is wanted by the driver.


Figure 8. Overall prediction error rate and the share of false positives at the overall error rate.

## VI. CONTEXT-AWARE AUTOMATION CASE STUDY

The use of comfort and infotainment functions in a car generally depends on the driver's preferences. In turn, those depend strongly on the intention for the drive, represented by *drive contexts*. As a consequence, the use of comfort functions depends on the drive context, e.g., *commuting drive* or *regular fitness club visit*. The class of climatic functions are a good example. For instance, while a driver heats up the car in the morning, lower temperatures may be wished-for after regular fitness center visits.

In the case of infotainment functions, e.g., integrated TV or audio player, this dependence is also visible. The selected audio source can be related to the driver's mood, time of the day, or again on the drive context. A driver may always listen to a certain radio station for traffic information on the way to work. But then, leaving work, she listens to CDs, music from the connected smartphone, or any other device connected. While this can have different reasons, the trip's goal or the intention behind the trip may be the most important. This leads to the idea of using the previously recognized drive contexts for audio source automation.

Starting from this idea and the presented drive context recognition, an explicit user preference model for infotainment functions can be realized. In the following, we will present an audio source selection automation algorithm. The algorithm will serve as a showcase for the usefulness of the drive context information in infotainment automation. Also, it will demonstrate a modular, purely probabilistic view for the use of an explicitly modeled user preference relying on contextual information.

### A. Entertainment source selection

In a modern car, there are several different audio sources to choose from. Typically, these include various types of radio sources, TV, internal storage and connected personal devices or a subset of these. The driver can select one of them at a time or disable all audio sources. Depending on the car's user interface, disabling audio is either done by disabling audio output or decreasing volume to zero.

The goal of the following case study is to showcase the use of the drive context recognition integrated into an automation of an infotainment function. The chosen function is the selection of the current entertainment source in the car. This means that we would like to recognize a driver's behavior in terms of selecting entertainment sources depending on the current drive context.

For the justification of declaring the *drive context* as the main context for entertainment source selection, we conducted a preliminary study. 30 subjects were interviewed and questioned about their user behavior in terms of infotainment systems. This also included the use of the navigation system, eliminating the need for the prediction of the current drive cluster related to the current drive context.

While 80% of the subjects always use the navigation system, 7% use it frequently and 13% use it occasionally, no one would never use it. This indicates the necessity of the prediction of the current drive cluster, hence the target of the current drive.

28 of the 30 use more than just one entertainment source. While two subjects would listen to the radio only, one of them depicted music as generally not important. The different sources used are listed in Figure 9.

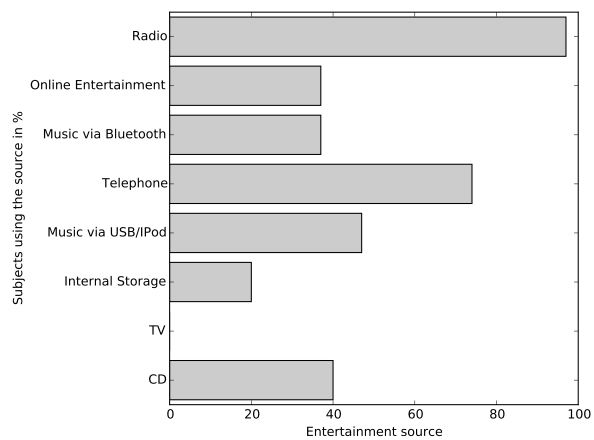



Figure 9. The amount of subjects using the different entertainment sources available.

25 of the 30 subjects stated to choose the entertainment source depending on the *drive context*. One answer was given, that because of the spare time drives mostly going abroad, the subject listened to the online entertainment rather than listening to the local radio. Another representative answer was that the subject listens to the radio to get to work and "moderately" start into the day, while listening to CDs or making calls on the way back home.

Three of the 30 subjects would never change their behavior according to the co-driver or the passengers in the back. The rest of the subjects would either give the control of the entertainment source to the passengers or completely go without any entertainment source. In Figure 10, additional influencing contexts were given.

The results from the interviews give some important indications about using the *drive context* for learning the driver's entertainment source behavior. The contextual information used in Section IV, daytime, number of passengers and the frequent drive clusters are the most important information for predicting the selected entertainment source. Therefore,

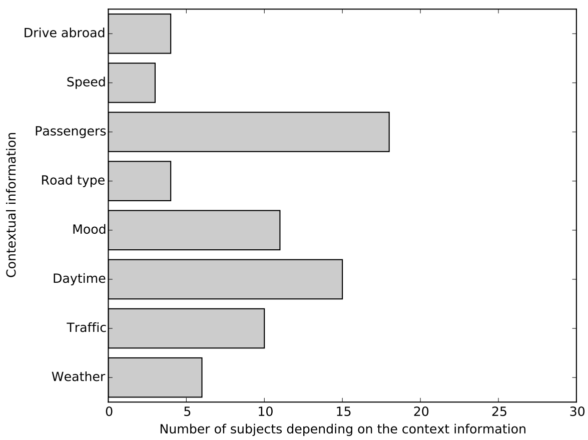



Figure 10. Additional dependencies influencing entertainment source selection.

the *drive clusters* abstraction delivers most of the information necessary for predicting the selected sources.

### B. Probabilistic view

This case study is targeted at automating a comfort function through imitating the driver's control behavior. In [20], a similar problem is formulated: Proactive recommendation or automatic activation of a certain camera-based comfort system at locations of interest. The locations are learned by observing a driver's individual use of the camera system. As an outcome, a modular system for location based activation of comfort functions is presented. It relies on a probabilistic view on the integration of abstract contextual information into the process of automation. This has several advantages over non-probabilistic methods, also discussed thoroughly in [20]. The main goal in [20] is formulated as

$$p(A|B) = \sum_{O} p(A, O|B) \tag{11}$$

being the probability of the driver intending to activate a comfort function under observation $B$ and learned locations $O$. With some basic assumptions, the probability of an intended activation $A$ can be simplified to

$$p(A|B) \approx p(A|O_j) \cdot p(O_j|B), \tag{12}$$

where $O_j$ is a context, describing an "abstract location". The observation $B$ can be any information currently accessible to the car. This separation of activation and location context in [20], makes it possible to model both contexts differently. In [20], this was used to implement user feedback and uncertainty quantification for better decision making.

This approach of separating the intention of an activation and the major influencing contexts of the decision can be adapted to our showcase. Instead of the probability of an activation $A$, we want the probability about the possibly selected audio sources. The selections of available audio sources will be denoted by $E = (E_n)_{n \in N}$, with $N = \{0, 1, \ldots, \sharp\text{sources} - 1\}$. Every $E_n$ represents a different audio source, e.g., radio, DVB-T or disc player. $E_0$ is not an active audio source itself, but implies *"disabled audio"* output of the entertainment system. This simple definition will be useful later on, enforcing

$$\sum_{i \in N} p(E_i) = 1. \tag{13}$$

This implies that we always want an answer for the automation mechanism.

The targeted automation does not depend on the notion of a learned location of interest, but rather on the current drive context. Therefore, the probability of an audio source $E_i$ being selected under individually learned drive contexts can be described as

$$\begin{aligned} p(E_i|B) &= \sum_{C} p(E_i, C|B) && (14)\\ &= \sum_{C} p(E_i|C, B) \cdot p(C|B). && (15) \end{aligned}$$

$C = (C_m)_{m \in M}$ is the family of observable drive contexts indexed by $M = \{0, 1, \ldots, \sharp\text{drive contexts} - 1\}$.

This is analogous to the situation in [20]. It is important to notice, that in [20], the probability of "no activation desired" is not calculated explicitly, but rather included in working with the probability of a wanted activation. In our case of automating the entertainment selection, the *"disabled audio"* selection is treated as another selection. Thus, the system always selects and activates a source.

In [20], some assumptions were made regarding the probability densities involved in equation (11). This allows the deduction of equation (12) for estimating $p(A|B)$. In our case study, similar assumptions can be made to estimate $p(E_i|B)$:

1) Given the information of the current drive context, any other information denoted by $B$ does not gain additional valuable information for the automation. This is inferred from our basic assumption that the drive context is the major dependence for the driver's audio source preference. This means all information of the observation $B$ is included in the context. This assumption induces the following simplification:

$$p(E_i|C, B) \approx p(E_i|C) \tag{16}$$

In the case of having more than one context, this can be a dangerous assumption, but is viable for the case study.

2) The previous evaluation of the presented drive context recognition in Section V-B shows a high accuracy. Taking the $k$ most likely drive contexts into account, the accuracy is close to $100\%$. If $I$ is the index set for the $k$ most likely selected audio sources,

$$\forall j \notin I : p(C_j|B) \approx 0 \tag{17}$$

can be seen as a viable assumption. Taking an even sharper condition, setting $k = 1$, the same assumption as in [20] can be made, leading to

$$\begin{aligned} \forall i \notin N \backslash I : \quad & p(C_i|B) \approx 0 && (18)\\ i \in I : \quad & p(C_j|B) \approx 1 && (19) \end{aligned}$$

for $C_j$ being the most likely drive context to be predicted. This approximation is supported by the high accuracy evaluated for only taking the most likely drive context.

Essentially, this simplifies the prediction of the audio source selection. Analogous to [20], the estimation of the probability for the selection of a specific source $E_i$ simplifies to

$$p(E_i|B) \approx p(E_i|C_j) \cdot p(C_j|B) \tag{20}$$

In the case of a recommendation system, the system may be free to offer the driver to play the audio source with the highest likelihood. If a source would not be available, the system should not recommend it and, therefore, does not have to recommend anything at all. But, in the case of a fully automated system, logic dictates to choose and play the most likely audio source available. If a source is unavailable, the source with the closest likelihood will be selected. The automatically selected source $E_{i_0}$ then is determined by

$$i_0 = \arg\max_{i \in I_{\text{avail}}} p(E_i|C_j) \cdot p(C_j|B) \tag{21}$$

where $I_{\text{avail}}$ is the index set for the available audio sources. As $p(C_j|B)$ is the probability of the current drive context, its calculation can be seen in Section V-B. From now on we will call it *Drive Context Distribution*. The probability distribution of the selected source, i.e., $p(E_i|C_j)$ will be shown in Section VI-C. The term will be called the *Source Distribution*.

Still, there is another simplification possible in equation (21). Since the $p(C_j|B)$ is a constant in equation (21), it can be eliminated, not changing the decision process on $E_{i_0}$. Thus, under given context drive $C_j$, the automated selection of the audio source can be formulated as

$$i_0 = \arg\max_{i \in I_{\text{avail}}} p(E_i|C_j) \tag{22}$$

When radio is selected as an audio channel, the preferred radio channel should also be automated depending on the drive context. Therefore, we define all radio channels to be denoted by $R = (R_k)_{k \in K}$ with $K$ being the index set for every listened or known radio channel. Analogue to the entertainment source selection, the most likely selected radio channel $R_{k_0}$ given a specified drive context $C_j$ can be formulated as

$$k_0 = \arg\max_{k} \arg p(R_k|C_j, E_1) \tag{23}$$

when $E_1$ denotes the radio source.

## C. Selected source

For the final decision, which source must be selected by the presented system, *Source Distribution* $p(E_i|C_j)$ must be learned. This can be done in several ways, involving to "observe" the active source on a trip. We will present two intuitive approaches to decide which entertainment source is considered to be *"listened to"*. One will serve as example for a whole class of point evaluation methods, while the other is motivated differently. As the same ideas for finding the active entertainment source apply to the active radio source, it will be referred to as *active source of a trip* from now on. These two basic ideas are illustrated in Figure 11. Figure 11a illustrates the time-lines for three different trips from a drive cluster. The colors orange and blue imply the relative time of one of the two active sources along each trips time line.

The first approach is to observe the activated source at a predefined point in time, relative to the drive's beginning or ending and declare it the *active source of the trip*. In Figure 11b, the first source that is observed as being active at the beginning of the trip is declared to be the *active source of the trip*. Defining a point in time, to designate the currently selected entertainment source as the *active source of the trip* is difficult. While the targeted automation should work at the beginning of the trip, it is hardly a good idea to do so. Most automotive audio systems will start when starting the car and choose to play the last active source or radio station. If the driver now has a different preference, the automation system must recognize this. Defining when the driver has settled for an active source is also non-trivial, because the driver may change the source periodically. This is illustrated in the middle trip in Figure 11a.

The second approach is to decide the *active source* depending on the source switching behavior from the driver. Periodic changes of the *active source* may strongly depend on other influences from the environment and the content of the audio source itself. In Figure 11a, the top trip shows some switching between the two sources. While the orange source may be solely for entertainment purposes, the second one may be preferred by the driver as news or traffic information source. The driver then would switch in between sources when this information is wanted, e.g., when stuck in traffic. Taking the longest active source is acceptable, as the driver listened to it for the longest, thus the audio source delivering most of the preferred content to the driver.

As for the sake of practicability of this case study, the latter approach was used. The focus is on the demonstration of the drive context for entertainment source selection.

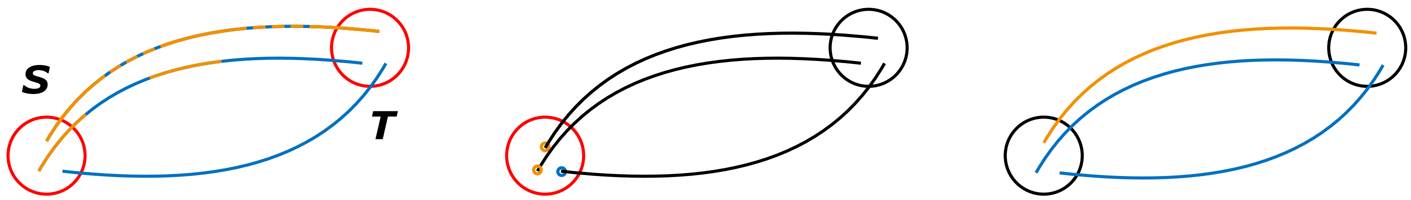
## D. Case study data

For this case study of drive context for personalized entertainment source prediction, eight drivers participated in a short-term study. The drivers were provided with prepared cars, logging the standard bus systems and the central entertainment system. The logging system was a prototype explicitly developed for this study and had to be installed in the cars with connections to the internal data bus system of the central entertainment system. The cars were provided for about four to six days to each participating subject, being enough time to recognize the working time. This provided the data for an offline evaluation of the algorithms. The entertainment system could not be controlled externally, making testing the automation online impossible.

The drivers were also interviewed, allowing the comparison of the offline prediction and the subjects statements. This would ensure a higher expressiveness of the short-term results of the study.

Gathering enough data from the subjects over this short time span was not possible with every subject. For the previously presented drive context algorithm to work, at least several drives for every context must be observed. The $MinDrives$, declared in Section III-A, was set to $MinDrives = 3$ for the GPS route clustering, as it was proved to work accurately with small values for $MinDrives$ (see Section V). Also, the predicting algorithm for the source selection automation needs some observations of every drive context.

In the case of this study, only six subjects delivered enough data for a significant analysis. This is enough to give a coarse




(a) Two sources on three routes

(b) Active source at the beginning

(c) Longest active sources per route

Figure 11. Two approaches for tagging the *active source of a trip*: 11a illustrates an example for a drive context's current active source (orange and blue) on time scales from **S**tart to **T**arget. 11b and 11c demonstrate declaring either one of the listened sources or the longest listened source of a trip as the trip's observed active source.

quality indication for the usefulness of the drive context in automation and the integration for the entertainment source prediction.

### *E. Evaluation*

For the evaluation of the in-field case study, the logs of the 6 subjects were analyzed. The analysis includes the clustering of frequent drives and subsequent recognition of the drive context. On top of the recognized context, the presented algorithm from Section VI-B was used to learn the subjects' current audio listening preference. The outcome of the comparison of the predicted information and the subjects' interviews are listed in Table I.

In the interview, the subjects were asked about their personal preferences of infotainment use. This included the dependence on the frequent drives, passengers, co-passengers and mood. The given information was useful for a better understanding of the preferences learned in the study. The information of the preferred audio sources itself was divided into three different categories of drive context: *home to work*, *work to home* and *non-work related / leisure* drives.

Table I was structured for ease of comparison between the prediction and the data from the interviews. While on the leftmost side are the numbers of 6 useful subjects, on the right side are the most three categories of detected drive contexts. Per category of drive context, the table includes the prediction of the presented automation algorithm, as well as the answer from the interview. The table shows the information given in the interview, while on the right, the algorithm estimated the user's preference.

A general problem was that the non-work related drives were not properly detected. This is due to the short-term of the study, as well as the time of the recordings. The cars were provided mostly over working days, while most non-work related drives are done at weekends. Nevertheless, Table I, showing the detected preferences and the information given at the interviews, indicates promising results.

The trivial case of a constant preference, non-dependent on drive context or any contextual information must not be a problem for the algorithm from Section VI-B. Subject number 4 never changes the active audio source and, therefore, is a good example for trivial preferences. The presented algorithm indeed has no problem recognizing the constant behavior and predicts radio and the channel properly and verifiable.

Subject 3 also shows the working of the preference recognition. The subject said it would not listen to different radio channels, thus, recognizing the radio as the preferred source was the primary target. The presented recognition always chooses the most likely radio channel as explained in Section VI-C. To stabilize the automation in a long-term study, the presented learning of entertainment source preferences has to implement a form of uncertainty quantification as shown in [20]. Also, the preference recognition does not take into account the use the telephone, because calls are not necessarily made for the same reason as audio source are selected. Therefore, *CD* was the best prediction result possible on work to home drives.

Subject 5 provided 4 days of logged information, mostly on workdays. Despite the small amount of data gathered, the prediction showed a significant difference in preference on work travel. While listening to CDs in the morning work travel, listening to the radio driving back home was clear and coincided with the data given at the interview. The non-work drives did not deliver enough information for a significant statement, but gave a coarse direction. The given influencing factors confirmed the use of the indicated information in Figure 10.

The presented case study showed that the presented drive context recognition works well as a basis for an explicit user preference model. Using the number of passengers and daytime information in the drive contexts is clearly a benefit as shown in the evaluation of the study. The influencing factors given at the interviews approve the benefits. This confirms the initial assumption, that the drive context is a major dependency of driver preferences in comfort functions and useful for recommendation and automation systems.

## VII. Conclusion and Future Work

In this article, we investigated the detection and prediction of frequent drive contexts as an important building block for automatized vehicle personalization. We proposed two different spatial clustering approaches for identifying frequent drive patterns in a GPS data set. The route independent Start-Stop-Clustering is promising, as it detects patterns independently of the chosen routes. The presented Bayesian Net's accuracy in differentiating frequent drive contexts was about 89% respectively 97% for a top 3 match.

We also presented a case study incorporating the recognized drive context. The study showcased the usefulness of the presented drive context recognition algorithm, when

TABLE I. Offline evaluation example and interview comparison.

| Subject | Home to work | | Work to home | | Non-work related | | Influencing factors |
|---|---|---|---|---|---|---|---|
| | Driver | Predicted | Driver | Predicted | Driver | Predicted | |
| 1 | radio | DAB, $R_{1,0}$ | radio | DAB, $R_{1,1}$ | Bluetooth | - | daytime, co-driver |
| 2 | radio | DAB, $R_{2,0}$ | radio/telephone | AuxIn | mostly radio | DAB, $R_{2,0}$ | daytime, co-driver |
| 3 | radio | FM, $R_{3,0}$ | CD/telephone | CD | CD/radio | radio | daytime, mood, co-driver |
| 4 | radio | FM, $R_{4,0}$ | radio | FM, $R_{4,0}$ | radio | FM, $R_{4,0}$ | passengers |
| 5 | CD | CD | DAB, $R_{5,0}$ | DAB, $R_{5,1}$ | misc. | - | daytime, co-driver |
| 6 | radio | FM, $R_{6,0}$ | radio/telephone | FM, $R_{6,1}$ | radio | FM, $R_{6,2}$ | weather, road type, passengers |

learning driver preferences. The targeted comfort function was the entertainment source selection and proved to work very well. It also showed a modular way incorporating such context information. The next step should be a long-term study for significant confirmation of the results. Also, the number of contexts should be increased and techniques for reducing the burn-in phase of the preference learning should be investigated. This would include techniques ranging from uncertainty quantification to collaborative approaches.

The duration of the learning phase is critical for using learning systems in the field of comfort and infotainment. In this regard, the presented algorithms need to be evaluated with further field studies before being deployed in series production vehicles.

## ACKNOWLEDGMENT

The authors would like to thank the participants of the studies and BMW Group for providing the drive data and the equipment, including the test vehicles. All location and drive data was anonymized to ensure the participants' privacy.

## REFERENCES

[1] S. Haas, K. Wiesner, and T. C. Stone, "Car ride classification for drive context recognition," in MOBILITY 2014, The Fourth International Conference on Mobile Services, Resources, and Users, 2014, pp. 61–66.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in Handheld and ubiquitous computing. Springer, 1999, pp. 304–307.

[3] M. Bazire and P. Brzillon, "Understanding context before using it," in Modeling and Using Context, ser. Lecture Notes in Computer Science, A. Dey, B. Kokinov, D. Leake, and R. Turner, Eds. Springer Berlin Heidelberg, 2005, vol. 3554, pp. 29–40. [Online]. Available: http://dx.doi.org/10.1007/11508373_3

[4] G. D. Abowd et al., "Towards a better understanding of context and context-awareness," in Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, ser. HUC '99. London, UK, UK: Springer-Verlag, 1999, pp. 304–307.

[5] A. Schmidt, "Ubiquitous Computing - Computing in Context," Ph.D. dissertation, Lancaster University, November 2002.

[6] O. Mazhelis, "Real-time recognition of personal routes using instance-based learning," in IEEE Intelligent Vehicles Symposium (IV 2011), 2011, pp. 619–624.

[7] J. Froehlich and J. Krumm, "Route prediction from trip observations," in Proceedings of the Society of Automotive Engineers (SAE) 2008 World Congress, SAE Technical Paper 2008-01-0201, April 2008, pp. 1–13.

[8] D. Tiesyte and C. S. Jensen, "Similarity-based prediction of travel times for vehicles traveling on known routes," in Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ser. GIS '08. New York, NY, USA: ACM, 2008, pp. 14:1–14:10.

[9] A. Brilingaite and C. S. Jensen, "Online Route Prediction for Automotive Applications," in Proceedings of The 13th World Congress and Exhibition on Intelligent Transport Systems and Services (ITS 2006), London, October 2006, pp. 1–8.

[10] K. Torkkola, K. Zhang, H. Li, H. Zhang, C. Schreiner, and M. Gardner, "Traffic Advisories Based on Route Prediction," in Proceedings of Workshop on Mobile Interaction with the Real World, 2007, pp. 33–36.

[11] O. Mazhelis, I. Žliobaite, and M. Pechenizkiy, "Context-aware personal route recognition," in Proceedings of the 14th international conference on Discovery science, ser. DS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 221–235.

[12] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in Data Engineering, 2002. Proceedings. 18th International Conference on, 2002, pp. 673–684.

[13] K. Laasonen, "Route Prediction from Cellular Data," in Proceedings of the Workshop on Context-Awareness for Proactive Systems (CAPS). Helsinki, Finland: University Press, 2005, pp. 147–158.

[14] K. Tanaka, Y. Kishino, T. Terada, and S. Nishio, "A destination prediction method using driving contexts and trajectory for car navigation systems," in Proceedings of the 2009 ACM Symposium on Applied Computing, ser. SAC '09. New York, NY, USA: ACM, 2009, pp. 190–195.

[15] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," Personal Ubiquitous Comput., vol. 7, no. 5, Oct. 2003, pp. 275–286.

[16] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," in Ubicomp, 2006, pp. 243–260.

[17] C. Zhou, N. Bhatnagar, S. Shekhar, and L. Terveen, "Mining personally important places from gps tracks," in Data Engineering Workshop, 2007 IEEE 23rd International Conference on, 2007, pp. 517–526.

[18] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering Personally Meaningful Places: An Interactive Clustering Approach," ACM Trans. Inf. Syst., vol. 25, no. 3, July 2007.

[19] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," in Proceedings of the 2Nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, ser. WMASH '04. New York, NY, USA: ACM, 2004, pp. 110–118.

[20] T. Stone, O. Birth, A. Gensler, A. Huber, M. Jänicke, and B. Sick, "Location based learning of user behavior for proactive recommender systems in car comfort functions," in Proceedings of INFORMATIK 2014, GI-Edition - Lecture Notes in Informatics (LNI), 2014, pp. 2121–2132.

[21] J. B. Macqueen, "Some methods of classification and analysis of multivariate observations," in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in KDD, vol. 96, no. 34. AAAI Press, 1996, pp. 226–231.

[23] M. Lewandowsky and D. Winter, "Distance between sets," in Letters to nature. nature publishing group, 1971, pp. 34–35.

[24] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," ACM Transactions on Mathematics Software, vol. 3, no. 3, September 1977, pp. 209–226.

[25] R. W. Sinnott, "Virtues of the Haversine," Sky and Telescope, vol. 68, no. 2, 1984, pp. 159+.

[26] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods

in data mining: A survey," in Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS, H. J. Miller and J. Han, Eds. Taylor and Francis, 2001, pp. 201–231.

[27] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," Computers & Graphics, vol. 23, no. 6, 1999, pp. 893–901.

[28] D. W. Scott and S. R. Sain, "Multi-Dimensional Density Estimation". Amsterdam: Elsevier, 2004, pp. 229–263.