How's My Driving? A Spatio-Semantic Analysis of Driving Behavior with Smartphone Sensors

Dipyaman Banerjee, Nilanjan Banerjee(✉),
Dipanjan Chakraborty, Aakash Iyer, and Sumit Mittal

IBM Research - India, New Delhi, India
{dipyaban,nilanjba,cdipanjan,aakiyer1,sumittal}@in.ibm.com

Abstract. Road accident is one of the major reasons for loss of human lives, especially in developing nations with poor road infrastructure and a driver needs to constantly negotiate with several adverse conditions to ensure safety. In this paper, we study several such adverse conditions that are relevant to safe driving and propose a novel method for identifying them as well as characterizing driving behavior for such conditions. Experimental results reveal that our proposed methodology is promising and more flexible than prior work in this area. In particular, our prediction results reveal that our methodology is an aggressive one where most of the bad driving behaviors are determined at the cost of a few instances of good behavior being falsely characterized as bad ones.

Keywords: Smartphone · Driving behavior · Analytics

1 Introduction

Road accident has been one of the banes of modern civilization, taking a significant toll on human lives. In U.S.A., motor vehicle crashes have claimed 4,544 teens between the ages of 16 and 19 in the year of 2005 alone[1]. While the rapid increase in the number of vehicles on the road has contributed to this scary statistics, negligent and rash driving has been the chief contributor. There have been many efforts to bring in some order to the traffic scenario by either monitoring and managing overall traffic or monitoring individual driving behavior. However, most of these solutions are primarily designed for developed countries with organized traffic infrastructure and practices. The solutions, consequently, are rendered ineffective for developing nations where the setting is much more chaotic. The problem becomes more complex in such a setting where it is not only the driver's fault that is responsible for an accident, but infrastructure, or rather the lack of it, and surrounding environment also play a significant role. The fact that India has the highest traffic accident rate worldwide, with 3,42,309 deaths reported in 2008 alone[2], is a testimony to that.

[1] WISQARS: www.cdc.gov/injury/wisqars.

[2] National Crime Records Bureau record: http://ncrb.nic.in/adsi2008/accidental-deaths-08.pdf.

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 653–666, 2014.
DOI: 10.1007/978-3-319-11569-6_51

A large body of work on driving behavior analysis exploits data from In Vehicle Data Recorders (IVDR) [8,10]. IVDRs are installed in the vehicles to provide information about speed, acceleration, manoeuvres, etc. The first application of IVDR was the event data recorder that was designed based on the “black box” used in aircrafts and primarily recorded crash information. Later, the event data recorders evolved to include specialized sensors integrated with vehicles to provide sophisticated information (e.g., vehicle steering wheel operation information) on the manoeuvring of a vehicle that could be used for driving behavior analysis. For example, the SmartCar testbed described in [8] recognizes driver maneuvers at a tactical level by recording the car’s brake, gear, steering wheel angle, speed and acceleration throttle signals. Further, it maps these maneuvers to video signals that capture the contextual traffic, the driver’s head and the driver’s viewpoint. Such integrated IVDRs are not widely applicable across different types of vehicles. For instance, a solution based on steering wheel operation information would not be applicable to auto-rickshaws that ply in large numbers on Indian roads. To solve this problem, portable mobile units [6] were installed on vehicles to gather general information such as speed, acceleration, rotation, location, etc.

A significant body of work is emerging that uses smartphones of the drivers instead of the expensive, special-purpose mobile units. Paefgen et al. [9] evaluate a mobile application that assesses driving behavior based on in-vehicle acceleration measurements and gives corresponding feedback to drivers. Eren et al. [2] propose an approach to understand the driver behavior using smartphone sensors, more specifically, the accelerometer, gyroscope and the magnetometer. Using these sensors, they obtain position, speed, acceleration, deceleration and deflection angle sensory information and estimate commuting safety by statistically analyzing driver behavior. Johnson and Trivedi [3] propose a system that uses Dynamic Time Warping (DTW) and smartphone based sensor-fusion (accelerometer, gyroscope, magnetometer, GPS, video) to detect, recognize and record these actions without external processing. Use of smartphones to detect driving behavior has also been applied to motorcycles, where mobility and motion characteristics are different from that of a car. An example is [4], here the authors propose a system to comprehend a motorcycle’s behavior using the acceleration and gyroscope sensors on the smartphone.

The IVDRs are either integrated or mounted on the vehicles and hence are not flexible enough to be applicable across a variety of vehicles. Mobile units, including smartphones, are loosely coupled with the vehicles and hence are free from this problem. However, none of the smartphone based solutions take ambient context (road conditions, traffic, etc.) into consideration while determining driving behavior. In this paper, we bridge this gap by first identifying parameters which define a *spatio-semantic ambient driving context* and propose a method for classifying driving behavior based on *segmentation* of a given trajectory along these parameters utilizing smartphone sensor data. Our specific contributions can be summarized as follows:

– A methodology for detecting *driving events* to characterize driving behavior and the *ambient context* from smartphone sensors, *accelerometer, gyroscope and GPS*.
– A methodology for classification of "good" and "bad" driving behavior given an ambient context.
– Performance evaluation of our methodology in characterizing driving behaviors against ground truths provided by expert observers.

2 Problem Description and Motivation

Driving is a skill that heavily depends on how the driver negotiates with the physical environment. We define *ambient context* as the factors in physical environments that become relevant to a driver. We classify ambient context into two categories:

1. Static Context: This includes context attributes whose value remains invariant for a substantial amount of time and across multiple trips. Examples of static context can be turns on the trajectory, road conditions such as potholes, bumps, etc.
2. Dynamic Context: This includes parameters which changes frequently with time and can vary across trips along the same route. This may include presence of other drivers, their driving behavior, temporary obstacles, etc.

Both static and dynamic context play an important role in determining drive quality. For example, a skilful driver would negotiate a sharp turn with a rough surface or a pothole on the road with more control than an unskilled or rash driver resulting in a safe and better experience for the people within the car and on the road. The type of context varies spatially and temporally. Figure 1a, b and c shows a few examples of static pathological ambient contexts, while Fig. 1d shows a pathological dynamic ambient context on a typical Indian road in New Delhi.

(a) A pothole (b) A rough patch (c) A bump (d) A dynamic ambient context

Fig. 1. Examples of ambient context

While smartphone based static ambient context determination has been investigated in [7], dynamic contexts are far more difficult to measure and model compared to static contexts and require costlier set-ups like on-vehicle video cameras and data from other city-sensing infrastructure [11]. Due to logistics issues, we leave out dynamic ambient context determination from our study. As we shall see, there are significant challenges to construct a methodical solution towards determining rich drive profiles using static context itself.

Before we establish the critical role of ambient context in influencing driving behavior, let us define what we mean by *good* and *bad* driving behavior in the context of this work.

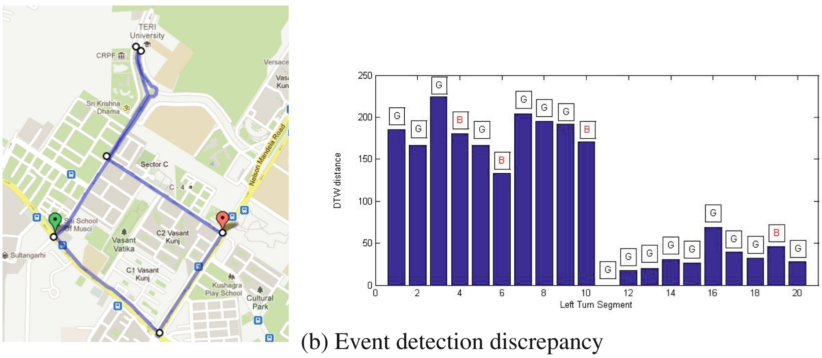
Definition 1. ***Good driving behavior:*** *– A "good" or "normal" behavior is where the driver negotiates an adverse condition such as a turn or a pothole, carefully, by driving with average speed profile for the adverse condition.*

Definition 2. ***Bad driving behavior:*** *– A "bad" or "rash" behavior is when the driver drives with higher-than-average speed profile for an adverse condition.*

We conducted an initial experiment to understand the impact of ambient context on driving behavior. We collected acceleration vectors of a car on two types of road surface while taking a turn: a smooth road and a road with a rough and bumpy surface. We requested the driver to drive in two states corresponding to *good* or *normal* behavior and *bad* or *rash* behavior on the same stretch of road (see Fig. 2a for the road stretch).

We implemented a prior method proposed by Eren et al. [2] to analyse driving behavior using smartphones. The method uses Dynamic Time Warping (DTW) to compare an unknown timeseries of drive data, with a few candidate classes. The assumption was, smaller the DTW distance, the better is the match between the test data and template. Figure 2b shows the results. DTW distances are plotted for each left turn, while being compared with a "good" left turn on a normal surface template in Fig. 2b. The actual labels are denoted by G and B representing good and bad driving respectively.

As can be clearly seen, the first 10 left turns have a very large DTW distance compared to the next 10 left turns. When compared against the true classifications, we observe some of the bad left turns actually have a better match to the



(a) Route for Experiments (b) Event detection discrepancy [G – Good Driving; B – Bad Driving]

Fig. 2. Demonstrating the relevance of ambient context

template than the good left turns, even though the template itself was for a good turn. Thus, ironically, on the rough surface, bad left turns matched better with a good left turn template than left turns which are truly good. This was caused by the inadvertent change in the speed profiles caused due to the roughness of the road surface. This highlights the need to consider 'context' when developing machine learning models for characterizing driving behavior, especially in chaotic driving conditions like those existing in India.

3 Solution Methodology

Determining driving context forms the basis of our work. As stated earlier, we only consider the static context attributes and determine how they are distributed along the driving trajectory. We observe that besides having the time-invariance property, static context attributes typically remain unchanged within a region in the trajectory e.g. rough patches, bumps, turns, etc. This allows us to divide the trajectory into segments (possibly overlapping) based on the attribute values. However, accuracy of such segmentation is critical as they represent context which directly impacts driving behavior. Table 1 lists the different static context attributes we consider in this work and some of the possible values they can have.

Table 1. Static contextual attributes

Road network	Straight, turns, roundabout, bends
Road neighborhood	School, traffic signal, market place, no label
Road surface condition	Smooth, bump, pothole

Road network information and road neighborhood information is available widely from Geographic Information Sources (GIS). Crowdsensed social media based data sources like OpenStreetMap[3] can augment the information quality available in GIS databases. However, there is no available large-scale data source that maintains road surface conditions. Hence, we try to estimate them using accelerometer readings from drive data across trips.

Figure 3 presents our solution methodology. Overall, our approach can be divided into three steps: (i) fusion of data from multiple information sources, including mobile GPS traces of the trip, to infer *spatio-semantically rich* sequence of segments, (ii) use accelerometer readings to augment the segments with road surface conditions like bumps, rough stretches, etc., and (iii) classify driving behavior in a given context. Next, we present the detail each of these steps.

[3] www.openstreetmap.org/

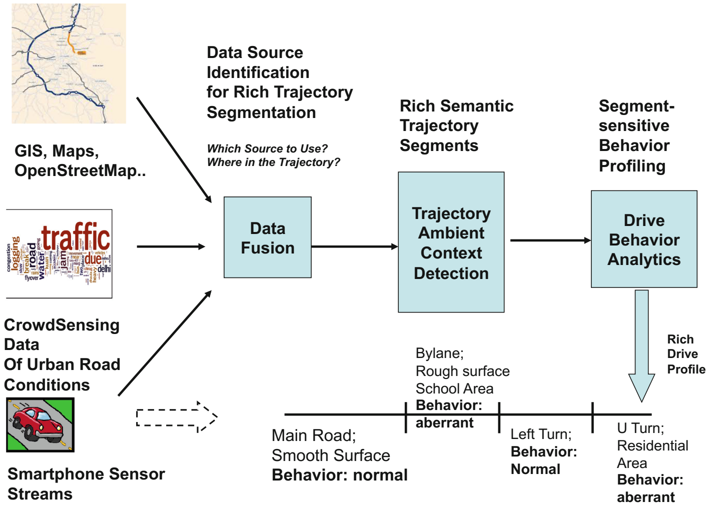


Fig. 3. Methodology for spatio-semantic analysis of driving behavior

3.1 Data Fusion

This step incrementally enriches the smartphone drive data to form a sequence of trajectory segments with corresponding static ambient context. Let T_i represent a trajectory of a single drive data from a trip i. $T_i = \{l_1^i, l_2^i, \cdots, l_n^i\}$ where l_j^i represent the j^{th} GPS observation in the trajectory of trip i. We take T_i and use standard map-matching algorithms to recover the corresponding road network data from a GIS database. Next, we use spatial join algorithms to recover the semantic region tags around each road segment. Semantic regions represent geographical areas like residential, commercial, etc. Subsequently, for each road segment, we perform a k-nearest neighbor query to recover the nearby points of interests (PoI). This provides us with a list of GIS objects for every segment from which we can recover nearby location artifacts like schools, market places, etc. and use them to tag each GPS observation associated with that segment. For example, we can denote $s(l_m^i) = \{\text{straight road}, \text{school}\}$; $s(l_n^j) = \{\text{turn}, \text{market}\}$; where $s(l_m^i)$ and $s(l_n^i)$ represent semantic information associated with the GPS points l_m^i and l_n^i, respectively. We perform a simple change-point detection on all such semantically enriched GPS data points present in the trip and mark a change point wherever there is a difference of at least one tag between two subsequent GPS points, i.e. $s(l_m^i) \neq s(l_{m+1}^i)$. This yields a sequence of homogeneous segments consisting of invariant tags fused from GIS sources. Let us denote any k^{th} segment as sem^k. We define $sem^k = \{l_m^i, l_n^i, [tags^k]\}$ where l_m^i, l_n^i are the segment boundaries and $[tags^k]$ denotes the set of semantic tags associated with all the points present in the segment.

3.2 Accelerometer-Based Segment Data Enrichment

The semantically enriched trajectory was seeded from GPS observations. However, they are not sufficient to monitor fine-grained road surface conditions. Specially, the sampling frequency of GPS may vary across trips and make any velocity-driven approximation of road surface conditions inaccurate with respect to segment boundaries. The accelerometer being a low-power sensor, it is acceptable to sample it with high frequency. We use the accelerometer readings from the trip to recover road surface conditions like bumps, rough stretches and augment the trajectory with that information.

An accelerometer records acceleration along 3 mutually perpendicular axes – x, y, z. In a horizontal orientation, one axis points towards gravity **g** vector. The phone can be in different orientations in a car. Wearable computing literature [5] has studied before that the **g** can be estimated reasonably well from an unknown orientation. This enables conversion of the readings to a fixed reference orientation plane where one of the axes points towards **g**. We use this to first map the raw data to our reference orientation plane. Let us call $z =$ the axis pointing towards gravity; $y =$ axis pointing towards lateral movement of the car; $x =$ axis pointing towards the forward movement of the car.

In order to improve reliability of predicting road surface segments while accommodating personalized variations in driving, we have designed a novel (1) cross-trip majority-voting driven segmentation method of drive data and (2) subsequent use of an adaptive DBSCAN algorithm, to arrive at rich segments specifying road surface conditions. The segments derived from data fusion is augmented with these to arrive at the final rich segments based on which the driving behavior will be evaluated.

Cross-Trip Majority Voting. The **z** axis acceleration component can be used to estimate the vertical disturbances a car goes through while driving and therefore can be used to detect rough surfaces as they typically cause the car to jump abruptly. However, in order to build a reliable segmentation of the persistent road surface conditions, it is important to perform a time series consolidation of the readings across multiple trips on the same road network because (1) the segments may be of varying lengths; (2) usage related interjections can add unnecessary false positives; (3) mechanical properties of the car and its age can add false positives; (4) unknown temporary obstacles (rampant on New Delhi roads) can add unnecessary false positives. We next discuss our segmentation methodology in detail.

Let $ACC^i = [z^i_1, z^i_2, ..., z^i_n]$ be n z-axis acceleration readings of trip i. Given m trips across all users, we first time-synchronize ACC^i with corresponding available GPS readings T_i and localize them. Wherever GPS readings are sparse, we use velocity-based estimation to interpolate the GPS points corresponding to the ACC readings. Thereafter, we cluster T_i by applying a criteria based on the deviation of the corresponding z^i_x value from the mean of observations across that trip: $\mu \pm r\sigma$ (where r is a threshold, μ and σ are respectively the mean and s.d. of ACC_i). We applied this simple criteria because major parts of the trip are expected to be having relatively smooth road conditions and rough patches and obstacles are interspersed by stretches of smooth surface. Since we would only like to differentiate between the values observed on a smooth surface vs. a rough surface, this timeseries clustering criteria is sufficient, though it is trivial to extend this to include multiple clusters. Let us denote such a cluster in the time series for trip i as L_i^m where $L_i^m \subseteq T_i$, and m is the cluster type. For detecting only smooth and rough surfaces $m = 2$. Next, we aggregate readings from multiple trips to find clusters that are repeating across trips and approximately in the same location using the following voting rule:

A point $l_i^x \in T_i$ provides a vote to another point $l_j^y \in T_j$ for it to be part of cluster m iff
(1) $euclidean\_distance(l_x^i, l_y^j) \leq \tau$;
(2) $l_i^x \in L_i^m$ and $l_j^y \in L_j^m$.

In other words, if two points from two different trips are sufficiently close and belong to the same cluster type in their respective trips, they will cast one vote to each other. For any point l_i^j, we define $c_m(l_i^j)$ to denote the number of votes l_i^j has received for cluster m. After aggregating the readings in this manner across all available trips, we can use the following algorithm to determine the list of points that belong to each cluster by using a threshold on the number of votes (τ_{vote}) a point has received. In this process, we find the list of points for which the conditions are invariant across trips, while reducing the false positives and non-persistent condition generated noise per trip. Algorithm 1 formally presents the algorithm.

Input: List of all trip points L_{in}, List of cluster types C, voting threshold τ_{vote}.
Output: List of trip points L_{out} annotated with cluster types
$L_{out} \leftarrow \emptyset$;
foreach ***point*** l_j^i ***in*** L_{in} **do**
  if $max_{m \in ST} c_m(l_j^i) > \tau_{vote}$ **then**
    $L_{out} \leftarrow (l_j^i, m)$;
  end
end

Algorithm 1. Segment Type Determination Method Based On Cross-Trip Voting.

Adaptive DBSCAN. The cross-trip majority voting method produces a list of points and their corresponding cluster types where the cluster type represents the road surface condition. Let $l_1, c_1, l_2, c_2, \cdots, l_n, c_n$ be such n majority-voted points and corresponding surface types along a road network. Our objective is to group these points such that the resultant timeseries indicates the segments corresponding to the surface conditions along the network. We employ a variant of DBSCAN, recently proposed in [1] on the output of the majority voting step to do this. We first explain why DBSCAN is not suitable for our problem, followed by the elaboration of our algorithm, where we modify [1] to suite our requirements.

For a set of points L, DBSCAN defines the density of l_i, denoted as $N_r(l_i)$, as the number of points that are present in a radius r around l_i. l_i is called a *core point* if $N_r(l_i) \geq MinPts$, where $MinPts$ is a user-defined constant. All the points around l_i present in a radius r are called *directly density-reachable* from l_i. A position l_j is *density-reachable* from l_i if there is a reachability relationship $(l_*)_1, \ldots, (l_*)_l$, where $(l_*)_1 = l_i$ and $(l_*)_l = l_j$. Two positions l_i and l_j are *density-connected* if they are both density-reachable from a core point l_c. Let L^β be a set of points, where $l_i \sqsubset L^\beta$ is density-connected with $l_j \sqsubset L^\beta$ for all $i \neq j$. We can randomly select the unclustered points and cluster all density-reachable points into the same cluster. This way, we can divide L into k clusters, where $0 \leq k \leq |L|/MinPts$. From our perspective, however, DBSCAN does not consider the surface types associated with each point. Thus, it is possible that DBSCAN would find clusters that only group nearby GPS points, without being sensitive to whether they represent a road segment consisting of static surface conditions. Moreover, density variations of the GPS readings can adversely affect the clustering. To accommodate these drawbacks, we introduce a further constraint to the clustering process.

Definition 3. *Gravity Segment-reachable.* *Two tuples l_i, c_i and l_j, c_j are gravity segment-reachable if position l_i is density-reachable from l_j and $c_j = c_i$*

Definition 4. *Gravity Segment-connected.* *Two tuples l_i, c_i and l_j, c_j are gravity segment-connected if l_i and l_j are model density-reachable from a core tuple l_{core}, c_{core}.*

This adaptive DBSCAN process performs the partitioning of the majority-voted L, C sequence to create partitions that represent road surface segments. From each such segment, we recover sem_k=l_m^k, l_n^k, c_{core}^k. The output from this step is augmented with the semantic segments produced during data fusion. The segments produced here have overlaps with the segments produced before. We run the change point detection once more to split the overlaps into a sequence of non-overlapping segments. For e.g. a $sem^k = l_m^k, l_n^k, [school]$ would be split into

$l_m^k, l_i^k, [school, road_{smooth}]$,
$l_i^k, l_j^k, [school, road_{bump}]$,
$l_j^k, l_n^k, [school, road_{smooth}]$.

3.3 Drive Profile Determination

Once we have created a rich segmentation of available trip data, our task is to measure driving behavior under the ambient conditions present in each segment. Let a trip T_i =$seg_1^i, seg_2^i, seg_3^i, \ldots, seg_N^i$ be N segments representing a road network $\mathcal{R}$. Let M=number of observed trips (across users) for road network $\mathcal{R}$. For a given seg_j^i, let $ACC_j^i[3]$ represent the vector of accelerometer readings in the jth segment of trip T_i. We capture this data and need to predict a driver's behavior in the segment. monitoring drive behavior [2] as collection of such fine-grained training data does not scale to practical settings. Our objective, on the other hand, is to use simple unsupervised clustering techniques to investigate whether it is possible to distinguish driving habits in each segment. For this purposent driving conditions. It is not realistic to use supervised approaches us, we attempt to distinguish drive habits over a certain ambient context into two classes : *normal* or *rash*. Given M trips and N segments on $\mathcal{R}$, we define a MxN matrix $\mathcal{A} = \lfloor a_{ij} \rfloor, 0 \leq i, j \leq M, N$, where $a_{i,j} = ACC^i_j$.

Segment-Sensitive Driving Behavior Clustering. Feature sets play an important role towards clustering quality of the data. Intuitively, acceleration variations of the vehicle is an indicator for goodness of behavior. This variation can be captured using the $ACCx^i_j$ component of ACC^i_j. However, different ambient contexts call for different boundaries to differentiate between good and bad driving behavior. E.g. speed profile in residential areas is different from speed profiles on bigger roads. Similarly, the $ACCz^i_j$ components can be indicative of the road surface induced shocks, which can give an estimate of how the driver is maneuvering different surface conditions. $ACCy^i_j$ on the other hand, provides lateral acceleration profile in turns.

We apply clustering techniques to cluster the data along each segment $\mathcal{A}_{*j}$ across multiple trips. This allows us to compare the driving behavior within the same segment across multiple trips. We transform the data in each cell a_{ij} to a set of features first. Subsequently, we apply the well-known K-NN clustering algorithm to find k clusters. In our case, $k = 2$. The clustering in K-NN is dependent on the choice of the seed(s). Our idea is to identify *normal* and *rash* behavior. Unfortunately, without training data, it is impossible to determine the exact data characteristics of these classes. Hence, we make use of a domain knowledge that most drivers usually drive in a respectful way and hence represent *median* behavior. We choose two seeds – one that is nearest to $\zeta(ACC_j)$, and one that is outside $\zeta(ACC_j) \pm r\sigma$ band, where ζ is $median(A_j)$ and σ is the standard deviation. After the clustering process, we label the set around $\zeta(ACC_j)$ as *normal* and the other cluster as *aberrant*. Note that many feature transformations (time and frequency domain) are possible on $\mathcal{A}_{*j}$ before the clustering step. In the experiments section, we present results with respect to a few well-adopted time domain features, and report our results with respect to our clustering approach.

4 Experiments

We have used a 3-month old Hyundai Santro car and a Samsung Galaxy II smartphone equipped with GPS, accelerometer and gyroscope to record data for our experimental evaluations. A sensing stack was developed on the Android platform of the smartphone to record data from each sensor with custom sampling rates. We secured the phone in one of the chambers in the car dashboard so that the forward movement of the car is recorded by the x-axis and the vertical movement by the z-axis of the accelerometer data. Understanding usage-related induced noise and other such human-generated disturbance to the data stream is out of scope of this paper.

A full-fledged rich data collection across many users for our problem is quite costly. Apart from the users needing to volunteer with time and money (fuel), most importantly, in absence of 'ground truth', it is hard to validate results on drive behavior. Hence, all our evaluations are based on a volunteer-driven data collected over 10 trips on a road stretch in New Delhi, shown in Fig. 2a.

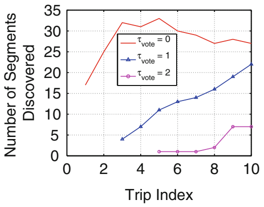
A single volunteer drove the car on the selected road segment in each of the trips. An expert driver in the passenger seat did an assessment of the drive quality (good or rash) for different segments and logged them as 'ground truth'. Note that according to the Definitions 1 and 2 given earlier, the ground truths should be determined using population based statistics, but in the absence of such statistics we have used subjective estimates of an expert driver as ground truths for this work. The accelerometer data was recorded with a sampling frequency of 30 Hz, while the GPS information was recorded every second. It took us approximately 10–15 min to record each trip. With this data, we focus on understanding: (1) Efficacy of our algorithms to identify ambient driving contexts (2) Performance of identifying drive behavior in those contexts[4]. Note that due to the nature of our methodology, there is no suitable reference algorithm to compare our results against. The most recent work [2] used dynamic time warping-based supervised classification. As described earlier, apart from conducting some initial experiments with this methodology and revealing its drawbacks, this methodology is unsuitable for our approach of developing an unsupervised learning methodology.

4.1 Ambient Context Determination

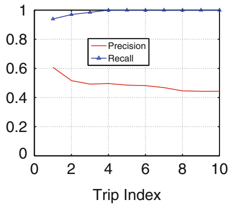
In our first experiment, we wish to understand the efficacy of identifying challenging ambient driving contexts. For this purpose, we have focused our experiments to a selection of ambient contexts involving turns and road surfaces to perform in-depth analysis of our methodology.

We determine two types of ambient contexts, *viz.*, turn contexts and road surface contexts. As mentioned, we employed GIS-based map-matching algorithm to the trajectory recorded by the GPS data for identifying the turns. Accelerometer based segment data enrichment and cross-trip majority voting approach described earlier were employed to identify segments in the trajectory of each trip that corresponds to the road surface conditions. We have reported the results for identifying bumps on our route (Fig. 2a). Three different values of τ_{vote} were used to demonstrate the efficacy of the cross-trip majority voting method. $\tau_{vote} = 0$ represents segments discovered from the union of all points across all trips which are labelled as 'rough'. Similarly, $\tau_{vote} = 1$ represents segments discovered from the points that are labelled as 'rough' by at least two trips and so on. Figure 4a shows the number (percentage) of bumps on the road that were discovered by our method. As we can observe, for $\tau_{vote} = 0$, initially, the number of segments discovered increases monotonically as we find new 'rough'

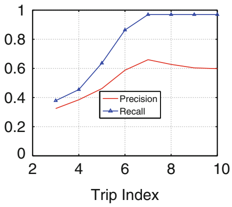
[4] In all our experiments we have used $r = 2.5$ for the clustering operation, both for segmentation as well as for driving behavior classification.



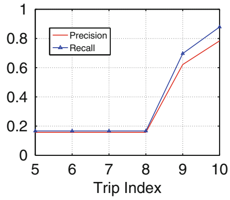
(a) Number of segments discovered



(b) Precision and recall for $\tau_{vote} = 0$



(c) Precision and recall for $\tau_{vote} = 1$

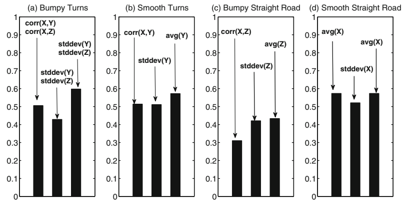


(d) Precision and recall for $\tau_{vote} = 2$

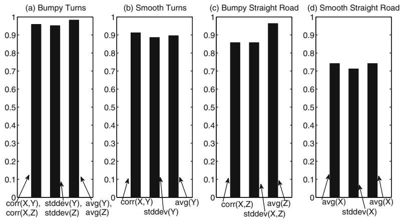
Fig. 4. Performance of ambient context discovery procedure

points on the trajectory as we include data from more trips, creating newer segments. However, increase in the number of segments also implies a corresponding decrease in the distance between two successive segment boundaries. Consequently, after a point, when the distance between two successive segments drops lower then the threshold of the adaptive DBSCAN algorithm, it automatically merges two successive segments resulting in a drop in the number of segments. This phenomenon can be observed from the plot shown for $\tau_{vote} = 0$. For higher values of τ_{vote}, increase in the number of segments is more sluggish as the probability for any point to acquire votes higher than τ_{vote} (and considered to be included in a segment) from same number of trips decreases with increasing τ_{vote}.

Figure 4b, c and d shows the accuracy of our segment discovery method which we compute by comparing against the ground truth. We observe that for all values of τ_{vote} the recall increases as more segments are discovered with increasing number of trips. The rate of increase becomes more sluggish for higher values of τ_{vote} as the segments are discovered slowly due to reason cited above. Observe that recall is highest for $\tau_{vote} = 0$ as it includes all points from all trips that are labelled as 'rough'. However, it also includes a large number of false positives which is reflected by the corresponding low precision value. For $\tau_{vote} = 1$, precision increases initially as we find more 'rough' points with increasing number of trips. However, as number of trip grows, after a point, it also starts including false positives resulting in a drop in its precision score. For $\tau_{vote} = 2$ we have a very low precision and recall at the beginning as it fails to find any segment for the reason stated above. However, due to the same reason it also includes minimum false positives as we increase the number of trips, resulting in a steady



(a) Precision for segment based localized clustering



(b) Recall for segment based clustering

Fig. 5. Driving profile determination

Table 2. Features used in clustering for the four ambient contexts we have considered

Ambient context	Feature 1	Feature 2	Feature 3
Bumpy turn	corr(X,Y), corr(X,Z)	stddev(Y), stddev(Z)	avg(Y), avg(Z)
Smooth turns	corr(X,Y)	stddev(Y)	avg(Y)
Bumpy straight road	corr(X,Z)	stddev(Z)	avg(Z)
Smooth straight road	avg(X)	stddev(X)	avg(X)

increase in the precision score. We therefore select $\tau_{vote} = 2$ as our operating threshold for the adaptive DBSCAN algorithm.

4.2 Driving Characterization

Next, we performed experiments to identify driving behavior for an ambient context and validated our results with that of the ground truth collected using the expert driver. Here, we evaluate our proposed method of determining driving characteristics given an ambient context. We have considered 4 different ambient contexts - bumpy turn, smooth turn, bumpy straight road, smooth straight road - to characterize bad driving behavior for each ambient context. The characterization has been done following two approaches: (i) clustering localized segments (segment at a given location in the trajectory) across all the 10 trips and (ii) clustering all segments corresponding to a particular ambient context. Table 2, shows the features sets we chose for clustering different ambient contexts we have considered. The choice of these features are influenced by the accelerometer axes, which is expected to show maximum variation under a given context as discussed before in Sect. 3. Figure 5a and b shows the precision and recall values for identifying aberrant driving behaviors with our approach for the 4 ambient contexts. Although the precision is not notably high, ranging mostly between 50–60%, the recall values are much higher (80–90%) than the precision values implying that our method is an aggressive method where most of the aberrant driving behaviors are identified at the cost of a few normal driving behaviors being characterized as aberrant. However, unlike prior supervised learning methods presented earlier in the motivation section, which are dependent on limited training set data, our methodology demonstrates the capability of recognizing the impact of circumstantial ambient contexts using a bottom-up, unsupervised analysis of driving behavior. We continue to experiment towards improving precision, while retaining the high recall values.

5 Conclusion

Recently, a few research efforts have investigated supervised methodologies for driving behavior detection using smartphone sensors. However, we argue that in chaotic road settings, *ambient driving context* assumes importance in categorizing the driving behavior of an individual and it is hard to practically implement supervised data driven methodologies. In this paper, we present an information fusion-based methodology for determining ambient driving context using multiple data sources including smartphone sensors followed by the determination of driving behavior for a given ambient context. We focus on static ambient context and experiment with real data collected for a 'chaotic' road stretch in New Delhi. We report results demonstrating the feasibility and utility of our approach, while presenting insights on achievable quality for the determination of such fine-grained spatio-semantic driving behavior.

References

1. Cartier, S., et al.: Condense: managing data in community-driven mobile geosensor networks. In: SECON 2012, pp. 515–523. IEEE (2012)
2. Eren, H., et al.: Estimating driving behavior by a smartphone. In: Intelligent Vehicles Symposium (IV), 2012, pp. 234–239 (2012)
3. Johnson, D.A., Trivedi, M.M.: Driving style recognition using a smartphone as a sensor platform. In: Intelligent Transportation Systems (ITSC), pp. 1609–1615 (2011)
4. Kamimura, T., et al.: A system to comprehend a motorcycle's behavior using the acceleration and gyro sensors on a smartphone. In: Global Research and Education (2012)
5. Kunze, K., et al.: Which way am i facing: inferring horizontal device orientation from an accelerometer signal. In: ISWC 2009, pp 49–50 (2009)
6. Mitrovic, D.: Reliable method for driving events recognition. IEEE Trans. Intell. Transp. Syst. **6**(2), 198–205 (2005)
7. Mohan, P., et al.: Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: SenSys 2008, pp. 323–336. ACM (2008)
8. Oliver, N., Pentland, A.P.: Graphical models for driver behavior recognition in a smartcar. In: Proceedings of the IEEE Intelligent Vehicles Symposium 2000 Cat No00TH8511, pp. 7–12 (2000)
9. Paefgen, J., et al.: Driving behavior analysis with smartphones: insights from a controlled field study. In: International Conference on Mobile and Ubiquitous Multimedia (2012)
10. Perez, A., et al.: Argos: an advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation. IEEE Trans. Intell. Transp. Syst. **11**(2), 463–473 (2010)
11. Qin, H., et al.: Heterogeneity-aware design for automatic detection of problematic road conditions. In: MASS 2011, pp. 252–261. IEEE (2011)