Moving Type Detection without Time Information

Martin Garbe*, Christian Bünning*, Anne Gutschmidt†, Clemens Cap*
*Department of Computer Science, University of Rostock, Rostock, Germany
Email: {martin.garbe, christian.buennig, clemens.cap}@informatik.uni- rostock.de
†Faculty for Economic and Social Sciences, University of Rostock, Rostock, Germany
Email: anne.gutschmidt@uni-rostock.de

Abstract—Today location technologies are integrated into many devices enabling location-based services. Movement data recorded with these devices can be uploaded to web sites and shared with others. Movement data can be organized using keywords and semantic tags, e.g. walking and running. Our main goal is to automatically classify movement data as walking, cycling or driving. In contrast to other work we use real world GPS data without time information. Users delete time information to protect their privacy. Over a period of two months we collected movement data from a popular track sharing platform and classified the GPS tracks using Decision Tree, Naive Bayes, Support Vector Machine and Naive Bayes Tree. Our results show that users can expect reasonable results from tag suggestion services detecting moving type even without time information. A second result of our work indicates that angle-based features are insignificant for classification using real world GPS tracks. Distance was the only significant feature in our study.

Index Terms—Data mining, classification algorithms, trajectory, angle-based feature, GPS

I. INTRODUCTION

The quality of location information produced by consumer devices rapidly raised in the last decade. This led to novel applications like location-based services (e.g. searching for restaurants nearby), community maps (e.g. OpenStreetMap [1]) and platforms for sharing movement data (e.g. users upload and share their trips and training tracks). Users recording their movement need a way to manage these tracks for later finding. This is comparable to photos which are saved in an album sorted by date, event or semantic tags. GPS tracks can also be organized using tags.

Intuitive tags for locations describe the location, e.g. Paris, and intuitive tags for moving data can describe movement type, e.g. biking. Automatic tag suggestion algorithms can help users to choose tags. Some research works [2]–[7] indicate that time and derived parameters like velocity and acceleration are necessary for good detection results.

But sometimes people do not want to give their raw movement data away including time information. For example people from OpenStreetMap project create maps and partly upload their moving data to prove that they visited the place they added or changed. This way people prevent any allegations of plagiarism. Some community members upload their movement data without time information. The reason for that are privacy concerns. Time information also contains not so obviously reasonable facts. One obvious fact reasoned can be that a person visited a place at a given time. A not so obvious fact reasoned can be that a person drove too fast on motorway.

Our main objective in this paper was to answer the question whether transportation mode detection is possible with reasonable results even without time information. Therefore we used real world GPS tracks from a web track sharing platform. Compared to other work [2]–[6] we had no influence on hardware devices or settings, like logging interval.

As a second objective we wanted to know which information hidden in GPS tracks is mainly responsible for good classification results. Therefore we made an in-depth analysis of the features and analyzed their influence on classification accuracy.

This paper is organized in the following way. Section II summarizes work in the field of movement type detection. Section III presents the data model and features we used for classification in detail. In Section IV we describe data collection, the classification task with four classifiers, two different feature selection schemes and the evaluation approach. Results are given in Section V including a discussion dealing with a surprising difference between our significant features and those found by others. Finally in Section VI we draw conclusions and give an outlook.

II. RELATED WORK

In [3] transportation routines were detected for modeling user behavior. To be able to differentiate between traveling by bus, foot and car, a hierarchical Markov Model was trained. A Rao-Blackwellized particle filter enabled predictions of traveler’s goals. The authors used map features to extract bus stops and to map movements to streets. A transition matrix represented transition probabilities between transportation modes. This way inference of transportation modes could be done. Our work differs in that [3] used map data and historical information of users. We had no further knowledge about users history and we were not working with maps.

An approach to recognize more activities using more sensors was done by Parkka et al. [4]. Besides GPS other sensors like accelerometers, thermometer and microphone were studied. The aim was to differentiate eight different activities (lying, sitting/standing, walking, Nordic walking, running, rowing with a rowing machine and cycling). Decision Trees and Artificial Neural Networks were used for classification. A similar work with many sensors was done by Ermes et al. [5]. Both work [4] and [5] needed many sensors to detect an activity. We only used GPS sensors which are available in most mobile phones nowadays.
The authors of [8] created a general framework for activity recognition. With the help of Relational Markov Networks a model for labeling locations and activities was created. After supervised learning the model could label locations as "AtHome", "AtWork", "Shopping", "DiningOut", "Visiting" and "Others". Only activities taking place in a fixed location were detected. In contrast to our study no moving types were detected.

In [9] and [6] GPS tracks of people were used to find a classification algorithm separating driving, walking, taking a bus and riding a bike. Calculations of change points split tracks into segments. Followed by a classification using Decision Tree, Baysian Net, Support Vector Machine and Conditional Random Fields to detect the transportation mode in every segment. For calculating change points time information is necessary. Therefore change points were not applicable in our work.

Another interesting approach is the sole use of GSM Cell IDs described in [10]. The advantage is that Cell ID information is always available. Besides power of the mobile device can be saved since no GPS is necessary. The authors showed that a distinction between moving and staying is possible with this sensor data. To make this moving detection possible a measurement of reachable Cell IDs at every location in the area is necessary. This is a very time-consuming task. Our goal was to not only detect moving and staying but to detect different types of moving.

A classification system using GPS sensors and accelerometers integrated in mobile phones was evaluated in [2]. The work separated stationary, walking, running, biking and motorized transportation modes. With a Decision Tree followed by a first-order discrete Hidden Markov Model a classification accuracy of 93.5% was achieved.

No accelerometers but GIS information supports classification in [11]. GPS data was combined with GIS features, like rail lines and bus locations, to differentiate train, bus, stationary, walk, car and bike mode. With Random Forest classifier an accuracy of 93.5% was achieved.

In [12] the authors used GPS movement data from vessels to detect different boat types. In this scenario special patterns such as "furthest distance from shore" could be used for classification. Using discriminant analysis, a classification model was created to group boats of type canoe, kayak, motorboat and sailboat.

III. MOVING TYPE DETECTION WITHOUT TIME INFORMATION

In this section we present our data model and features used. The calculation of features is discussed in detail.

A. Data Model

In general a trajectory $t$ consists of a sequence of locations $t = (l_1, l_2, ..., l_n)$ where $n$ is the number of locations. Movement is represented by a sequence of latitude, longitude and time triplets $l_x = (\text{lat}, \text{lon}, \text{time}).$ In our scenario explicit time information is missing which results in a sequence of tuples $l_x = (\text{lat}, \text{lon}).$ Some temporal information is implied by the order of locations.

In some cases it is possible to reconstruct time from movement data when recording interval is known. Reconstructing an unknown setting is a huge challenge since there is a great amount of possibilities to log positions. To mention only a few logging can be done by time interval, when getting away more then a given distance from last position or when changing the direction.

B. Features

In the literature many features were suggested to detect transportation modes. We were restricted by the lack of time information having a sequence of $(\text{lat}, \text{lon})$ tuples only. Hence the following features were used in our study: distance (Dist), mean turning angle (MTA) and heading change rate (HCR). In the following we discuss each feature in detail. For an explanation of $h_x, d_x, \alpha_x$ and $\beta_x$ see Figure 1 as an instance. The variable $n$ represents the number of locations of the trajectory.

1) Distance: Distance moved is represented by feature Dist. Other work [6] indicates that Dist is an indicator for movement type. It is computed by Equation (1).

$$\text{Dist} = \sum_{i=1}^{n-1} d_i$$

2) Mean Turning Angle: The feature mean turning angle (MTA) was used in [12] and describes how much the direction was changed in average. The needed turning angle was calculated by Equation (2).

$$\beta_i = |\alpha_i - \alpha_{i+1}|$$

Therefore the mean turning angle is presented by Equation (3).

$$\text{MTA} = \frac{\sum_{i=1}^{n-2} \beta_i}{n-2}$$

3) Heading Change Rate: Finally the feature heading change rate (HCR) was suggested by [6]. The parameter counts how often direction was changed. It is calculated by Equation (4).

$$\text{HCR} = |L_c|/\text{Dist}$$
with \( L_c \) as set of locations where turning angle exceeds a certain threshold \((\beta_c)\). The study of Zheng et al. [6] suggested a threshold of \(15^\circ\) to recognize a direction change. We chose the same threshold value. Hence if turning angle \(\beta_i\) exceeds \(15^\circ\) the location \(l_{i+1}\) is added to \(L_c\).

IV. EXPERIMENT

In this section data collection and preparation procedure is discussed. Some classification algorithms have additional requirements on data which is also discussed. Furthermore we evaluate steps to achieve best classification results and steps to extract the most significant features. Finally evaluation criteria are examined.

A. Data Collection and Preparation

For our study we collected data from an online track sharing platform called GPSies [13]. The motivation for choosing an online track sharing platform was to easily fetch real world moving data which is already tagged.

In a previous study [14], we analyzed how much trajectory data was uploaded to different sharing websites. GPSies was the one with most uploaded tracks in the period of two month (50,916 tracks) starting May 1st, 2010 until June 30th, 2010. At the beginning the RSS feed was grabbed every 5 minutes and parsed for new items. Afterwards the GPX tracks, which are referenced in RSS feed, were downloaded from the website. We decided to take these nearly two years old data because newer data is not supposed to be a better source. GPS technology in smart phones and GPS loggers did not advance significantly in the last two years.

Users of GPSies are manually classifying their tracks when data is uploaded. A small part of the data had multiple classes, so we filtered these since our aim was to compare different movement types and we needed tracks of single classes. Figure 2 depicts the ten most frequent categories and the corresponding number of GPS tracks uploaded and tagged in the two month period. Overall there were 28 categories whereas the 10 most famous categories were represented with 98.6% of the items. The other 18 categories represented only 1.4% of data.

The aim of our study was to compare popular moving type detection methods on real life community data. In the literature [2]–[6] very often the classes walking, cycling and driving by car are mentioned and analyzed. Assigning the categories of downloaded tracks, see Figure 2, into these classes we end up with walking = \{walking, hiking, nordic walking\}, cycling = \{racing cycle, biking tour, mountain bike\} and driving = \{car, motorbike\}. Only having three categories at the end is one limitation of choosing an existing track sharing platform and not carrying out an own study with people logging and tagging their movement.

The next step was cleaning up all data. While reading the GPX files with an XML parser a small amount of files had invalid XML and were not taken into account. For best possible representation of problem description a stratification was done. This ensures less error prone classification handling in the following steps. The class driving consists of 209 items and is therefore the smallest one compared to the others. This leads to 209 items of each class after data preparation.

To support the idea of Open Science [15] and reproducible results all data collected and used here can be downloaded from our Open Science Repository [16].

B. Classification Models

In the field of data mining many different classification algorithms are available. All algorithms have scenarios where they perform best. Generally there is no best classifier. For our classification task we applied Decision Tree (DT), Support Vector Machine (SVM) and Naive Bayes (NB) because these are often used classifiers. This enables comparison of results with other publications where similar moving data is analyzed. Additionally we applied a hybrid of Decision Tree and Naive Bayes classifier [17]. The Naive Bayes Tree (NBTree) is designed as Decision Tree with Naive Bayes classifiers at the leaf nodes. According to [18] this hybrid performs sometimes better than normal Decision Trees.

While training classifiers the problem of overfitting has to be prevented. Overfitting is recognized when classifiers learn training data perfectly and do not learn the general structure underlying the data. To prevent this phenomenon statistical cross-validation can be used. Hereby learning data is separated into training and validation data. We used 10-fold cross-validation which means all data is split into 10 data sets. In each of the ten passes nine sets are chosen as training and one as validation set to prevent overfitting.

Besides the problem of overfitting, which affects all classifiers, some classifiers make assumptions on data. When working with these classifiers the assumptions on the data have to be verified. One assumption for Naive Bayes classifier is that predictor variables are not highly correlated each other. This can be checked with a scatter plot matrix comparing all variables. Regarding to Figure 3 there is no strong linear correlation between the features Dist, HCR and MTA. The highest correlation 0.39 was calculated between Dist and HCR. This means all three parameters can be used for classification using Naive Bayes.
C. Feature Selection

The problem of selecting a relevant subset of features is called feature selection. Using this subset the best possible classification performance is achieved. For our study we used a feature selection algorithm called forward selection (FwS) [19]. Searching for a relevant feature subset is done by wrapping the whole classification process. The wrapper classifies all data using different combinations of feature subsets which results in a list of subsets and regarding classification performances. Finally the feature subset with best classification performance is presented as best classification result for the wrapped classifier.

Forward selection begins with an empty set of features followed by testing every single feature. The feature with most performance improvement is permanently added to the model and therefore added to the feature subset. In the next step a subset with two features is evaluated whereas in each step another feature is added to the subset. Again the subset with two features and best performance is remembered. The process continues until an aborting criterion is fulfilled.

Regarding our first question to find best classification of data we used the aborting criterion of "no more increase". This means no more features are added to the subset if the new features are not improving the classification performance anymore.

Our second question was to evaluate which features play the most significant role and which attributes might be insignificant. For answering this question the aborting criterion "no more significant increase" was chosen. This results in a subset of features where every feature increased classification performance significantly. For all classification tasks Rapid-Miner [20] was used.

D. Evaluation Approach

To evaluate classifiers we use accuracy defined by Equation (5)

\[ ACC = \frac{c}{N} \]  

whereas \( c \) is the number of correctly classified items and \( N \) is the number of all items. This is similar to Accuracy by Segment \( (A_S) \) as mentioned in [6]. These segments were defined as tracks of single transportation mode. Applied to our data this means every track is a segment and therefore \( ACC = A_S \). An alternative accuracy metric \( A_D \) was also defined by the authors of [6]. It is calculated by Equation (6).

\[ A_D = \frac{\sum_{j=0}^{c} \text{CorrectSegment}[j].Dist}{\sum_{i=0}^{N} \text{Segment}[i].Dist} \]  

This means \( A_D \) is calculated as sum of distance of all correctly classified tracks divided by the sum of distance of all tracks. The metric \( A_D \) represents the accuracy in relation to overall distance of all moving data. A classifier optimizing \( A_D \) being applied to data set \( X \) results in best classification of longest tracks. In our scenario short tracks should be treated equally to long tracks. Therefore we use \( ACC \) which takes number of tracks into account.

V. RESULTS

A. Transportation Mode Detection Accuracy

First we will present results for best overall classification. The aim was to detect the semantic transportation mode with all features. In the second part results using only significant features are given.

1) Overall Accuracy: In Table I, II, III and IV confusion matrices of Decision Tree, Naive Bayes, Support Vector Machine and Naive Bayes Tree are presented. The tables show how many items could correctly be classified.

One observation from the tables is that walking tracks are occasionally incorrectly classified as driving but the converse is not the case. The confusion matrices also show that mostly walking tracks are correctly classified followed by biking and driving tracks.

The overall accuracy of all tested classifiers is presented in Figure 4. Best classification (73.35%) was obtained by using Naive Bayes Tree. The second best accuracy (72.55%) was computed using Decision Tree whereas the difference between the first and second is only 0.8%. Support Vector Machine achieves 69.53% and Naive Bayes 68.08%. The combination of two classifiers (NBTree) performed better than Decision Tree or Naive Bayes alone.
2) Significant Features: In the last section the overall performances of different classifiers were calculated and compared. The feature selection step was done by using forward selection scheme with the appropriate classifier as inner operator. Surprisingly different features were finally chosen by the forward selection scheme. The selection ended if no additional feature improved classification performance. Naive Bay Tree chose Dist, MTA and HCR for final classification whereas Naive Bayes chose Dist combined with MTA. Support Vector Machine only chose Dist and Decision Tree chose Dist and MTA feature. This raised the question whether all features are significant and necessary for classification.

Investigating this phenomenon another feature selection was done. The forward selection used so far was configured to stop trying new features when there is no more increase in accuracy.

We changed this parameter to stop when no more significant increase occurs. Therefore at the end of forward selection only significant features should be selected.

In Figure 5 the classification accuracy is shown comparing the two feature selection settings. In contrast to other research work the two features MTA and HCR have no significant influence on separating classes. Each forward selection chose only the Dist attribute whereas the other attributes were discarded. Best classification (71.13%) was obtained by using Naive Bayes Tree followed by Decision Tree (70.49%), Support Vector Machine (69.53%) and Naive Bayes (67.94%). Figure 6 is supporting this observation. A classification with HCR or MTA as single feature results in poor accuracy.

B. Discussion

We used real world data from a track sharing website to detect transportation modes. Moving data came without time information. In the following we will discuss our data source and classification with significant features.

1) Data Source: In general time is very important for detecting moving type because many parameters used for classification depend on velocity. Classification of Zheng et al. resulted in 75% accuracy with four categories (walking, driving, biking, going by bus). We managed an accuracy of 73% with only three categories (walking, biking, driving).
Estimating Zheng et al. had only three categories their classification accuracy would be surely above 80%. They have a better classification but also have time information which supports their task. Due to data source items of category bus driving were not available in our study.

2) Significant Features: For our study we chose features which were often used by others. Besides distance moved, mean turning angle and heading change rate were calculated. Our study shows that Dist is most useful to separate classes. Surprisingly MTA and HCR do not add any significant value for classification task. One possible explanation could be the inhomogeneous device settings. In contrast to other research work we had no influence on recording interval, GPS hardware or any other parameter. This out of lab work shows what can be accomplished with real world data.

Another reason for the inaccuracy of MTA and HCR maybe lies in the real world tagging behavior. In carefully designed studies users tag their data precisely. Even if they change transportation mode for a very short time it is tagged. As a result the track would be tagged by transportation mode used most of the time. Furthermore start and end of recording do not begin exactly with the tagged movement. For example the user starts recording with GPS receiver at home, walks to the car and then drives away. Finally all was tagged as driving. This behavior adds noise to the data but represents real world behavior.

VI. CONCLUSION & OUTLOOK

A. Conclusion

In this paper we studied how accurate the semantic transportation mode can be detected from real world movement data without time information. The absence of time information is intended when users want to publish movement data but do not want to publish all time constrains for privacy reasons. The objective was to determine whether these users can still use tag suggestion services for transportation mode or is time information urgent necessary for that.

Our results show that a classification accuracy of 73.35% is achievable using Naive Bayes Tree. This means tag suggestion services are able to find tags for describing the movement type with a reasonable accuracy. Furthermore our results indicate that angle-based features for classification do not have significant influence in classification accuracy taking unknown recording intervals of real world devices into account. As a consequence track sharing platforms working with movement data without time information only need to consider the distance attribute for classification. This saves much computational costs when preparing data.

B. Outlook

One limitation of our work is that no map data can be used for calculating map features as suggested by [3] and [11]. Nowadays free map data is available including much details, e.g. OpenStreetMap. Extending the feature list used in our work with suggested map features will most probably result in a better classification accuracy.

ACKNOWLEDGMENT

Christian Bünning was supported by a grant of the German National Foundation (DFG), Graduate School 1424, Multi-modal Smart Appliance Ensembles for Mobile Applications (MuSAMA).

REFERENCES
