QROWD - Because Big Data Integration is Humanly Possible

Innovation action

D6.1 – Real-time inductive analysis

Author/s	Luis Paulo Faina Garcia and Patrick Westphal
Due date	30.11.2017
Delivery date	01.12.2017
Version	0.2
Dissemination level	PU
Status	Final

Table of contents

ABSTRACT

Increasing the citizens mobility is one of the topics that involve Smart Cities and has become a very important subject of government agendas in the last years. This includes predicting traffic jams, decreasing the travel time of the citizens, predicting parking space available in the city center or even improving the routes between places to avoid accidents on roads and streets. However, collecting the data, process this information and also reuse crowd feedback from the citizens is a challenging task. In this deliverable we describe the design decisions and overall architecture of the QROWD analytics component based on the use case of deriving a user's transportation mode from sensor data captured on the user's mobile phone, that stems from the use case BC2#UC1 detailed in Deliverable 2.2. After describing the available data sources, we explain necessary preprocessing steps and discuss promising machine learning techniques for this task. Besides these more technical considerations, we also show where and how crowd feedback could improve the performance of the analytics component.

EXECUTIVE SUMMARY

This deliverable describes the overall architecture of the QROWD analytics component performing the task of detecting a user's transportation modes given sensor data captured on the user's mobile phone, based on the available data sources, its characteristics, and circumstances that influenced design decisions on the current architecture.

The deliverable serves as a technical documentation for developers involved in implementing the analytics component of the QROWD infrastructure, and for persons contributing to other work packages sharing interfaces to this component. Besides this, it specifies the overall approach of how we are tackling the problem of deriving transportation modes from sensor data which could be of interest to other researchers.

As part of the technical issues, we will present our ideas of how and where Data Mining, Machine Learning techniques and crowd feedback can be incorporated into the process to improve the performance of the transportation modes prediction and provide accurate analysis about the mobility over smart cities using sensor data.

This deliverable presents the initial design decisions of the analytics component and initial results of techniques already implemented and will be complemented by the deliverable D6.3 focusing on details of the spatio-temporal analytics.

1. INTRODUCTION

In this deliverable we give an overview of the main components used in the analytics component of the overall QROWD infrastructure. The design decisions are based on the business use case BC2-UC#1 ‘Modal Split’ as described in deliverable D2.2 ‘Business case requirements and design’. This use case covers the computation of the percentage of travellers using a particular means of transportation, and turned out to be a fundamental formal metric to understand the citizens’ habits in terms of traveling and commutation. These insights are crucial to get feedback with regard to former operations performed to optimize the traffic infrastructure of the Municipality of Trento, and to derive new steps for future optimizations. Since the sensor data available on the Modal Split use case is not provided as real-time data stream, we did not consider real-time analysis so far. Real-time data will have to be taken into account in future deliverables in WP6.

Whereas former approaches to collect modal split information through citizen surveys were costly, time consuming and only performed twice each decade, the aim of the QROWD project is to gain those insights with lower costs, in a higher frequency and in a similar quality, based on an active collaboration of the municipality, the citizens of Trento and industry partners. In terms of the analytics part of this goal we consider commutation information provided by the Trento citizens, data concerning the traffic infrastructure of Trento provided by the municipality, and road network, and traffic data from map and navigation products of the company TomTom. The main task here is to detect a person’s transportation mode based on sensor data gathered on their mobile phone.

Due the sparse labeling scheme of the training data we had to go beyond pure inductive analysis techniques and also consider further mechanisms to derive a suitable training set. Besides these more technical considerations, we will describe how the respective analysis tasks can be extended using crowd feedback to improve the prediction performance.

This deliverable is structured as follows: Section 2 introduces the data to be used by the analytics components. In Section 3 we describe and motivate the hybrid learning architecture of our approach and discuss individual steps in the overall workflow. Section 4 touches on the prediction part and shows options to improve the performance by means of crowd feedback. Section 5 concludes this deliverable.

2. DATA SOURCES

In this section we introduce the datasets available for the Modal Split use case as described in deliverable D2.2 ‘Business case requirements and design’. As mentioned above, the main task of this use case is to gain insights to the citizens’ habits in terms of the used transportation modes in the Trento area. Other than making use of paper-based surveys, these information should be derived from data available to the project consortium, allowing to provide the modal split information in a higher frequency, with lower costs and similar accuracy.

2.1 i-Log

The i-Log mobile application was designed by the ƧMAЯTRAMS laboratory[1] from the Trento University to collect sensor data from smartphones, also allowing complementary user feedback. The captured information of main interest for deriving transportation modes are the GPS position, the acceleration, as well as the orientation and angular velocity measured by gyroscope sensors. In the context of QROWD, i-Log will be used as the communication channel with citizens, gathering their data, and enabling the opportunity of asking citizens to verify that the analysis performed on the collected data is accurate.

In a first trial, the i-Log app was used by a cohort of students to record sensor data of their movements in the city of Trento, as well as requesting additional feedback, as for example their current transportation mode. However, this feedback information was gathered only in fixed intervals of 30 minutes and thus does not provide a complete labeling of all their movements.

Figure 1 depicts an example of these data showing the GPS track of one i-Log user during one day of tracking. One can clearly spot three main regions with higher probability of presence labeled with the letters A, B, and C. The regions A and B are areas with low population density and suggests places like the user's home or workplace whereas C is the city center that might have been visited for free time activities.

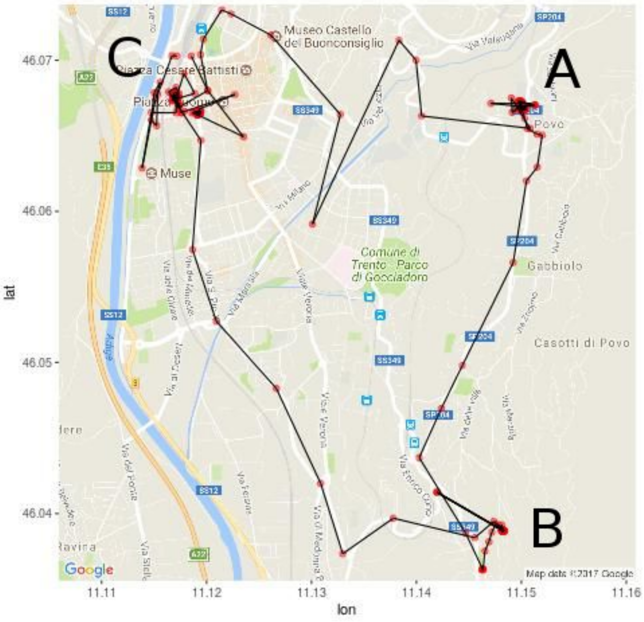

Figure 1: Position of one i-Log user during one day

[1] http://trams.disi.unitn.it/

Using only location data for learning transportation mode classifiers is highly discouraged because of the noise level which could deprive the information and the low number of samples per minute. Also depending on the means of transportation (e.g. trains or buses) and position (e.g. inside buildings) there might be no GPS coordinates at all due to a bad signal quality. The accelerometer and gyroscope are the most informative and most stable features and different literature sources showed that the prevalent transportation modes (car, bicycle, trains, metro and walking) could be reliably distinguished and predicted by means of each of those sensors.

2.2 LinkedGeoData

LinkedGeoData (LGD) [1] is an open data project providing an 'RDF mirror' of the OpenStreetMap [2] project that is interlinked with other prominent RDF data sources like DBpedia and GeoNames. The LGD data can be queried via a SPARQL endpoint and REST API, or downloaded as release compilations.

Following the main data model of OpenStreetMap the geospatial information contained in LGD is expressed in terms of *nodes*, *ways* (being a collection of nodes) and further characteristics modeled as classes. So in case a certain node in OpenStreetMap had the additional information attached that it represents a bus stop, this node resource will become an instance of the class *BusStop* in LGD's RDF representation.

The LGD dataset will be used as geospatial background knowledge for learning symbolic classifiers (see Sec. 3.3). Here the structured information, e.g. expressing that a certain GPS coordinate of a user's GPS trace recorded by the i-Log app is close to a bus stop, or on a railway line, may give additional hints for a more robust classification.

2.3 TomTom MultiNet/MultiNet POI

MultiNet and MultiNet POI are two commercial products provided by TomTom covering a wide range of geographical and topological features. In particular MultiNet and MultiNet POI model very detailed and accurate descriptions of road traffic related aspects and points of interest, respectively. The data is provided in different Geographic Data File Format [3] and Shapefile format [4] serializations.

The main primitives of MultiNet and MultiNet POI are *points*, *lines*, *areas* and *relationships* between them. More complex map constructs are built by means of those primitives and may be further described through a comprehensive set of feature or service categories, like 'Bus Stop', 'Railway Line', 'Open Parking Area' etc.

Like the LGD dataset, TomTom's MultiNet data can be used as structured background knowledge for the symbolic learning algorithms ran by the DL-Learner framework. Since the DL-Learner requires the background knowledge to be in RDF format, a transformation step is needed.

2.4 Bike Sharing Data

The Municipality of Trento provides structured background information that can be used for learning transportation mode classifiers. One example of such background information is data about bike sharing stations. The provided datasets contain information about the position as well as the number of bikes available at a bike sharing station. So, whenever a user's trip starts or ends at such a station this could be an indicator that a shared bike was used.

The respective datasets were introduced in the deliverable D4.1 'Data Catalog' under the dataset IDs 1014 and 1027.

2.5 Transport Data

Further datasets concerning the infrastructure for biking in Trento were compiled on the municipality and province level. They contain the geo locations of bike lanes and cycle paths. These are valuable information that can be exploited by map matching techniques as explained in Section 3. So whenever a user's GPS trace can be mapped to one of those cycle paths with a sufficient confidence a bike might have been used for transportation.

The corresponding datasets are listed in the deliverable D4.1 'Data Catalog' under the IDs 1016, 1019 and 1043.

2.6 Parking Data

Data sources that provide the geo locations of places that usually serve as a starting or ending point of car trips were compiled by the Community of Trento and introduced as datasets 1017, 1018, 1022, 1024, 1052, 1054 in the dataset catalogue deliverable D4.1. Besides the actual polygon descriptions of the parking area further information e.g. about the parking fees, the opening hours and so on are contained.

2.7 Data Concerning Services and Facilities Around the City of Trento

As already mentioned, the LinkedGeoData dataset provides a rich set of points of interest which are of importance especially for the symbolic learning approach. However, since LinkedGeoData is a user-driven collection of geo information the completeness in terms of mapped points of interest might differ across the considered regions of Trento. Thus, we also consider datasets describing services and facilities around the city of Trento as provided by the municipality and explained under the dataset IDs 1026, 1028, 1029, 1030, 1031, 1032 in deliverable D4.1.

2.8 Other datasets

Before having received access to the i-Log and TomTom MultiNet data, we did a literature review and started experimenting with different datasets available in public data repositories. Besides data describing city traffic flow we used in initial experiments for traffic jam prediction, we focused on accelerometer data for activity recognition. The respective dataset is available in the UCI Machine Learning Repository[2] and was compiled for human activity recognition using accelerometer information. It contains data from a wearable accelerometer mounted on the chest collected from 15 participants performing different activities like working on a computer, walking, standing, going up/down stairs, talking while standing etc. This data has similar information to the data collected by the i-Log mobile application and can thus provide useful insights about the performance of different machine learning techniques. We preprocessed the dataset and removed 4 classes not related to transportation modes. The final data had 3 predictive attributes related to the accelerometer axis and 3 classes: standing, walking and going up/down stairs. The dataset was mainly used for initial evaluations to find promising techniques for the statistical machine learning part (see Section 3.2).

As we already mentioned in Section 2.1 the i-Log data is not perfectly labeled through user feedback. Thus, to be able to apply inductive learning techniques deriving classifiers from example data, we need to add missing labels and find interchange points. To support this *initial classifiers* based on accurately labeled data are needed (see Figure 2). To get this labeled data we performed self tracking experiments using another mobile app called Sensor Log[3] which allows capturing arbitrary mobile phone sensor data and storing it with a user defined label. The corresponding data is stored in an SQLite database and can be exported to the CSV format. Classifiers based on this data were mainly trained with statistical machine learning techniques.

3. HYBRID LEARNING

This section introduces the foundations of the QROWD analytics component based on Data Mining and Machine Learning to solve the task of providing Trento-wide modal split statistics considering citizen feedback, data from the Municipality of Trento and industry partners, as described in D2.2 ‘Business case requirements and design’.

Considering the given datasets the main source for deriving the transportation mode is the sensor data from the i-Log app. This sensor information is purely numeric rather suggesting statistical learning methods. However we argue that the vast amount of structured geospatial background data available should help to improve the prediction quality. Thus we also want to apply symbolic machine learning

[2] https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer

[3] https://play.google.com/store/apps/details?id=com.hfalan.activitylog&hl=en

techniques working on RDF data. To also make use of the numeric sensor data in the symbolic learning part we plan to convert information derived in a preprocessing step (described in Section 3.1) to RDF. A further route we plan to pursue is to make use of map matching techniques [5] which may provide semantic annotations to a user's movements.

The overall approach is a hybrid ensemble learning framework and the resulting architecture is depicted in Figure 2. We consider four main parts: Preprocessing techniques described in Section 3.1, learning methods described in Section 3.2 and 3.3, the classification part introduced in Section 4, and crowdsourcing services provided by the work package 3, described in the Sections 3 and 4. On the left hand side of Figure 2 one can see the data sources used to train a set of statistical and symbolic classifiers. However, before the actual training is performed pre-processing steps are required, as illustrated in the figure. The set of classifiers is applied in the prediction phase, as shown on the right hand side of Figure 2. Here the input is a collection of sensor data from i-Log, which is again pre-processed and then used for classification. If needed requests to crowd workers can be performed through the crowdsourcing services as shown on the bottom.

3.1 Preprocessing

To make better use of the accelerometer or gyroscope data during learning it is important to transform the data in a preprocessing step removing the temporal dependency and adding the frequency spectrum. Moreover in this step the sensor data series is split into chunks such that each chunk only represents one single transportation mode.

3.1.1 Splitting

To find 'split points' where the transportation mode changed in the overall (unlabeled) sensor data series one could simply make use of a user's velocity. However depending on the sensors available in the user's mobile phone a velocity value might not be reliably derivable. Besides this, splitting whenever the velocity drops to zero might cause many, very small sub-sections.

Another option would be to make use of some additional classifiers we call *initial classifiers* based on gyroscope or acceleration sensor data that could predict the major transportation modes (car, bus, train, metro, tram, bike, walking). Using sliding window techniques on the sensor data series these initial classifiers should always return one prevalent transportation mode. Whenever this changes and the prediction gets ambiguous this can be considered as an interchange point. This ambiguity can be measured using entropy models on the multi class prediction values. To give an overly simplified example one could consider the situation where 'car' is predicted clearly for one window: car: 1, bus: 0, train: 0, tram: 0, bike: 0, walking: 0. If the situation changes to: car: 0.16, bus: 0.16, train: 0.16, tram: 0.16, bike: 0.16, walking: 0.16, one could consider this as a split point.

Besides finding split points where the transportation mode changed, another preprocessing task is to distinguish movements that contribute to the overall modal split and movements made e.g. while a user is working in an office. One obvious hint would be if there is no movement at all, e.g. when the user is sleeping. Another way to detect these ‘non-movement’ periods would be to make use of entropy models as described above or train an additional initial classifier e.g. for office work. Besides this, the available background data could be used. If a user’s current GPS coordinate points to an area labeled e.g. as ‘building’ this could also be used as an indicator.

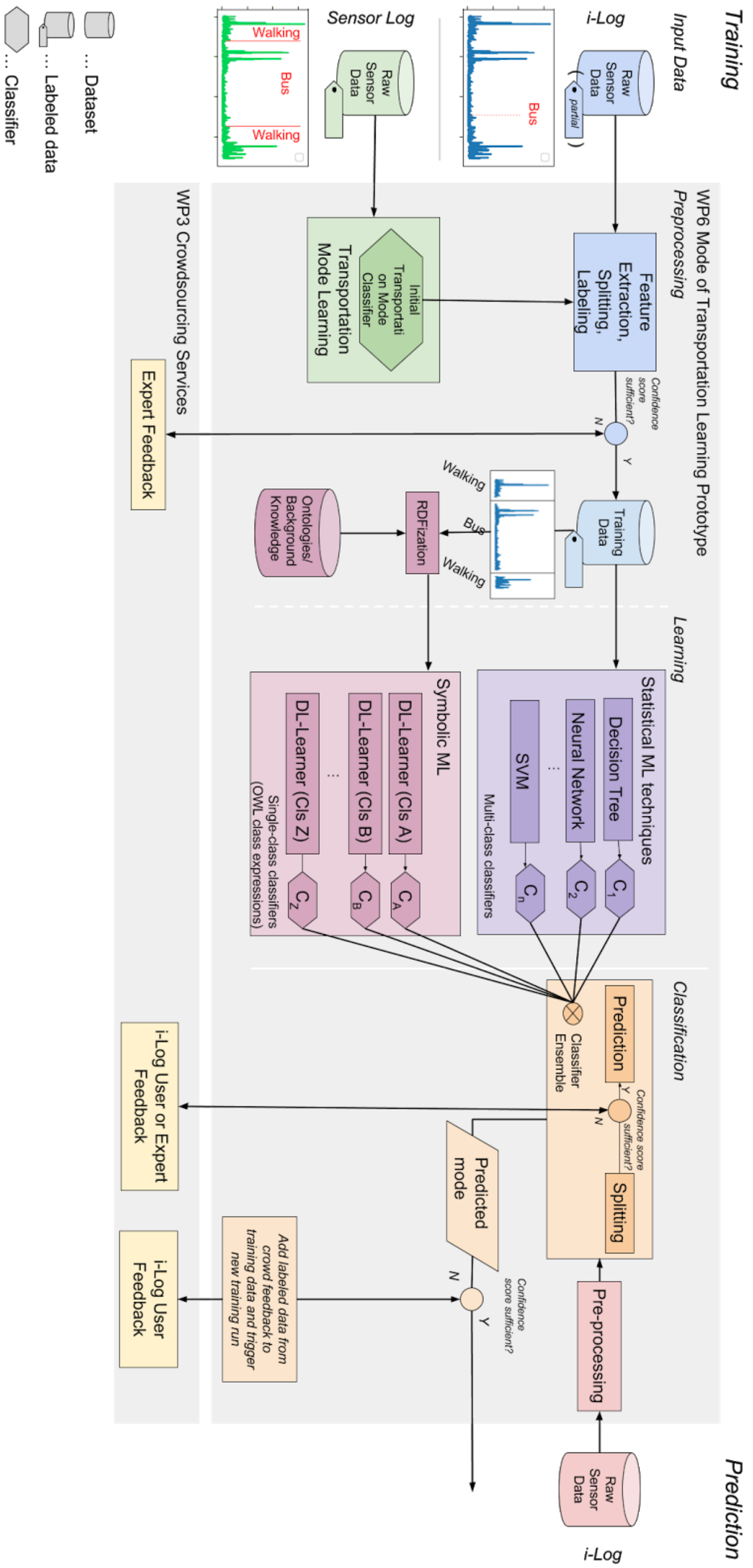

Figure 2: Flowchart with the evaluation, prediction and crowd feedback steps.

In case the confidence values of the respective outcomes for split points or the overall trip detection are too low we plan a feedback channel to ask an analytics expert for a decision. This expert feedback would only be needed during the preprocessing of the training phase which should ideally be run only once.

3.1.2 Feature Extraction

The next step is the transformation of the sensor data series into the frequency spectrum. For that we can use Fourier or Wavelet transformation. In the end of the preprocessing step we get a vector representing the characteristic portions of different frequency bins. Figure 3 shows an example of these data over time and frequency for 2 activities: walking and standing.

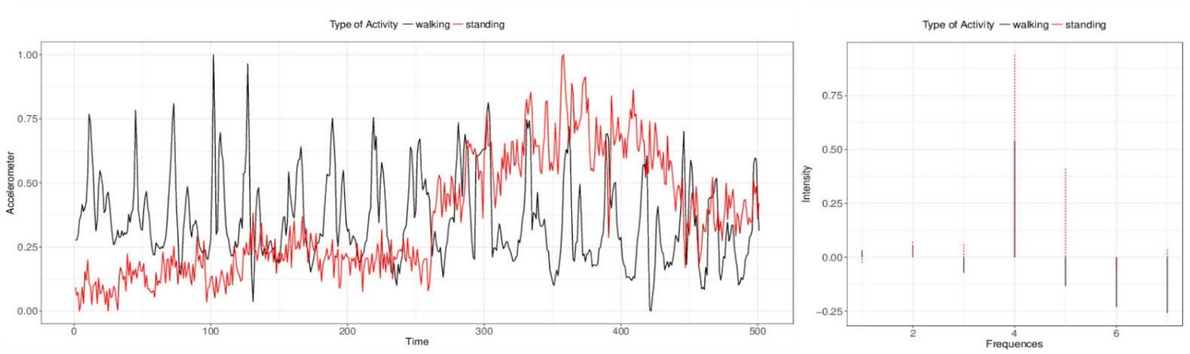

Figure 3: Accelerometer data for 2 activities over time and frequency.

Such vectors derived from the now labeled i-Log sensor data is used to train and validate different classifiers. Whereas the statistical machine learning techniques can be applied directly on these vectors, they need to be translated to RDF for the symbolic learning algorithms. One straightforward way to do this, is using a resource representing the measurement of one such window and assign RDF properties for each frequency bin, e.g. frequency40Hz, frequency60Hz etc. Another option would be to transform this vector into an RDF Datacube[4].

3.2 Statistical Machine Learning

The most promising statistical machine learning techniques we found during our experiments with the human activity recognition dataset from the UCI Machine Learning Repository described in Section 2.8 comprised Decision Trees (DT) [6], k-Nearest Neighbour (kNN) [7], Naive Bayes (NB) [8], Multilayer Perceptron (MLP) [9], AdaBoost [10] and Support Vector Machine (SVM) [11]. The results achieved on this dataset after the preprocessing steps described in the previous section are discussed in the following.

The concrete algorithms used in this evaluation were Decision Trees (DT) with Gini index with a minimum number of samples to split an internal node equal 2, k-Nearest

[4] https://www.w3.org/TR/vocab-data-cube/

Neighbour (kNN) with k=5 using Euclidian Distance, Naive Bayes (NB) with Gaussian likelihood function, Multilayer Perceptron (MLP) with one hidden layer with 100 neurons, AdaBoost, Support Vector Machine (SVM) with a radial kernel and RandomForest (RF) with 10 DTs. All the algorithms are validated using 10-fold cross-validation with default parameters.

Figure 4 shows the average performance of the 7 classifiers for different sizes of static windows. The accuracy of the classifiers decreased or remained constant when the window size was increased. The best algorithms were Random Forest and Decision Tree. The best window size was 500 and 1000 points. With 500 points, the Random Forest achieved around 80% of accuracy.

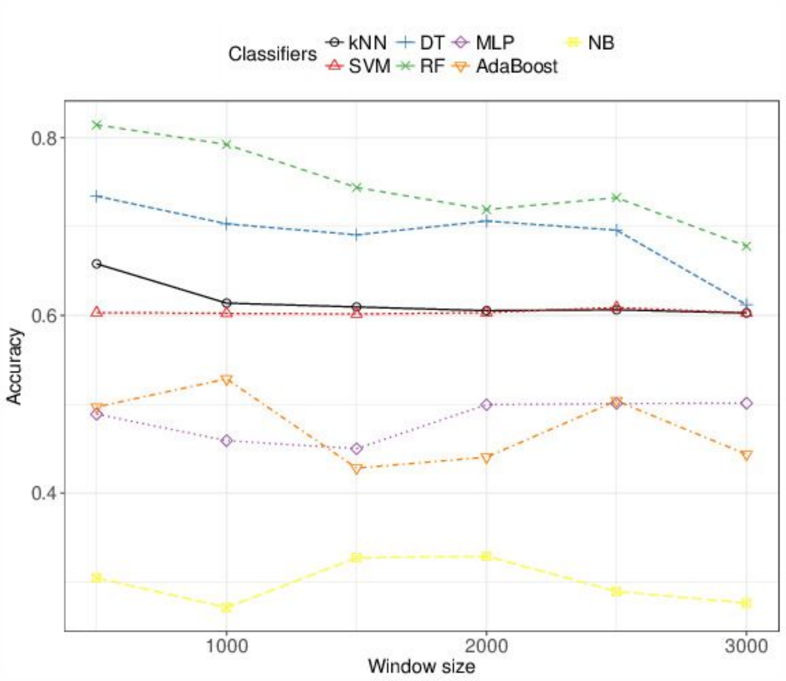

Figure 4: Accuracy of classifiers with different time windows

In future steps we will re-run this evaluation on the actual i-Log sensor data, improve the windowing algorithms and consider hyperparameter tuning and feature selection for further optimization.

3.3 Symbolic Machine Learning

In the symbolic learning part we apply the DL-Learner [12] which is a framework for supervised machine learning algorithms working on OWL, RDF and Description Logics. Other than the techniques discussed in the previous section these algorithms rather learn symbolic descriptions of structures that distinguish a set of positive examples from a set of negative examples. Considering the vast amount of structured background data the application of the DL-Learner framework seems promising.

The symbolic classifiers learned by the algorithms of the DL-Learner express the distinguishing parts of the underlying structured data for e.g. taking a bus compared to all other transportation modes. A (made-up) example of such an expression for the bus case could look like

(hasWindow some (frequency40Hz some double[> "0.7"^^double]))
and (hasWayPoint some (isNear some BusStation))

which could be verbalized as “something that has an evaluation window with the 40 Hz frequency bin above 0.7 and at least one of its waypoints is close to a bus station”.

4. PREDICTION

This section describes to the part of deriving a user’s mode of transportation for a given collection of sensor data points, considering background knowledge and classifiers learned during the learning phase. These predictions are the building blocks for the computation of the overall modal split which gives a Trento-wide overview of the used transportation modes.

The raw input sensor data is again provided by the i-Log application. To apply the classifiers from Section 3 the same preprocessing steps have to be applied first. (The details are omitted in Figure 2 for clarity.)

In the actual prediction phase each of the classifiers will generate a prediction and a confidence score reflecting how reliable the prediction is. In case of a low confidence score, a crowd feedback step will be performed which will be detailed in Section 4.1. All the individual predictions can then be checked against some heuristics, reflecting ‘usual’ usage of transportation modes. An obvious example of such a heuristic would be that it is quite likely that the usage of motorized transportation means is usually preceded and followed by a walking phase, e.g. when walking to the bus stop etc. Further heuristics would cover the typical lengths of a trip, e.g. that it is rather unlikely to take a bus for just 5 seconds.

4.1 Crowd Feedback

In the overall architecture sketch in Figure 2 there are three cases where crowd feedback could improve the learning performance. The first case may occur in the training phase and was already covered in Section 3.1.1. Here experts are asked to decide about the start and end points of trips and its interchange points. In a corresponding crowd worker task a curve similar to the plot on the left hand side in Figure 3 could be shown. The expert could then manually set the requested points on this plot.

The remaining two cases concern the prediction phase. Here the idea is to directly ask the corresponding user about the transportation modes used in the recent past exploring the historical movements of the same user (mobility patterns of the same citizen) and perhaps using the extra datasets described in Section 2.5, 2.6 and 2.7 to increase the confidence score.

Another option is to present a map showing the low confidence split points and ask the user for confirmation. To provide the necessary context information the map should also show previous sub-traces that could be detected and labeled reliably and nearby points of interest like e.g. bus stops. If the velocity can be derived from the given sensor data we plan to additionally show velocity information at the respective waypoints to give further hints.

Related to this kind of user feedback is the case where we could detect the split points reliably but only got bad confidence scores for the actual predicted transportation mode. Here we plan to show the i-Log user a similar map, but this time just asking to select a label for the sub-trace in question. To gain new insights from this user feedback the newly labeled sample will be added to the training data and a re-training run will be scheduled.

5. CONCLUSIONS

In this deliverable we presented the architecture of the analysis component performing the task of detecting a user’s transportation modes considering sensor data captured on the user’s mobile phone, and background information describing the traffic infrastructure and further geo characteristics of the Trento region. The information about which means of transportation were used by a Trento citizen is the main building block for the Trento-wide modal split statistics which give valuable feedback about the citizens’ habits in terms of traveling and commutation.

We described available sensor data and background information about the Trento region, and motivated design decisions for the QROWD analytics component. We further showed ways to improve the performance of the overall task by means of crowd feedback.

6. REFERENCES

[1] Stadler, Claus, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web* 3, no. 4 (2012): 333-354

[2] Haklay, Mordechai, and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing* 7, no. 4 (2008): 12-18.

[3] ERTICO - ITS Europe. GDF - Geographic Data Files, available at https://web.archive.org/web/20140920023431/http://www.ertico.com/gdf-geographic-data-files

[4] Environmental Systems Research Institute. ESRI Shapefile Technical Description (1998), available at http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

[5] Yan, Zhixian, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. "Semantic trajectories: Mobility data computation and annotation." *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, no. 3 (2013): 49.

[6] C4.5 Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1):81-106.

[7] k-NN Mitchell, T. M. (1997). Machine Learning. McGraw Hill series in computer science. McGraw Hill.

[8] Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In 10th European Conference on Machine Learning (ECML), pag. 4-15

[9] Haykin, S. (1999) Neural Networks - A Comprehensive Foundation. Prentice-Hall

[10] Freund, Y. & Schapire, R. E. (1995) A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting

[11] SVM Vapnik, V. N. (1995). The nature of Statistical learning theory. Springer-Verlag.

[12] Bühmann, Lorenz, Jens Lehmann, and Patrick Westphal. "DL-Learner—A framework for inductive learning on the Semantic Web." *Web Semantics: Science, Services and Agents on the World Wide Web* 39 (2016): 15-24.