Towards an automatic diary: an activity recognition from data collected by a mobile phone

Rudolf Kadlec and Cyril Brom
Charles University in Prague
Faculty of Mathematics and Physics
Czech Republic
rudolf.kadlec@gmail.com, brom@ksvi.mff.cuni.cz

Abstract

We present our initial work on an “automatic diary” recognizing and storing episodes of human daily activities. The goal is to create an application that will perform activity recognition based on data collected from a mobile phone. This includes a GPS location, WiFi and Bluetooth signals. Our aim is to combine these sensory data with information from publicly available databases of points of interest thus identifying restaurants, schools etc. Until now we have collected several months of hierarchically annotated data, created a desktop viewer of the logged data and experimented with inference of activities on two different levels of abstraction. Several machine learning algorithms were tested in these experiments.

1 Introduction

Human activity recognition is a tool that enables wide range of possible applications for a healthy lifestyle [Consolvo *et al.*, 2008] , helping elderly people [Kröse *et al.*, 2008] etc. Activity recognition also fits well into the context of lifelogging [Bell and Gemmell, 2009] – continuous logging of all possible information related to a person’s life. Decreasing prices of storage capacity enables us to continuously store many details of our daily lifes. We can store our location, photographs and even audio at an acceptable price. The key question is how to make the log accessible to humans. So far we can search it by time, or use full text search on recorded audio [Vemuri *et al.*, 2006]. We think that an automatic activity recognition can add a valuable key that can be used to search the lifelog in various applications.

The idea of lifelogging is fueled by current advances in mobile technologies. Smart mobile phones provide a wide range of sensors that can be used for human activity recognition. Thanks to extended battery life time it is now possible to continuously store information from GPS, WiFi, Bluetooth and accelerometer almost for a whole day. Imagine that just by wearing your Android mobile phone a summary of your daily activity could be automatically computed for you. During the day values from mobile phone sensors will be logged and at the end of the day the computer will present you several possible explanations of your todays activity. Among these explanations you will pick the one that best matches what you really did. Then you can share this information with your friends or family through Facebook or any other social networking service. Without any effort you will get statistics showing how and where you spend your time, these statistics can help you improve managing your time in the future. With the use of your diary you will be also able to better recall old episodes, e.g. a medieval castle visit last year. Recall of this episode will immediately show similar episodes of your life just like YouTube shows similar videos to the one you are just watching.

In this paper we approach the goal of enriching the lifelog by experimenting with the activity recognition on two levels of abstraction. The first is the level of atomic activities like sleep, work, watch TV etc. The second is the level of higher activities like visiting school, shopping, training etc.

The rest of the paper continues by describing the hierarchical activity representation used in our application. Then we detail the architecture of our system and procedure for collecting data. Further we review related works in the field of activity recognition. After that we will show two machine learning experiments, the first experiment focuses on low level activity recognition while the second deals with classification of longer time periods.

2 Activity representation

Findings from psychology suggest that people often perceive activities in a hierarchical way [Zacks and Swallow, 2007], e.g. an episode *Work day* can consist of a *Commute*, *Work* and again *Commute* episodes, where the *Commute* episodes can be further decomposed into *Walk*, *Travel by bus*, *Waiting at a bus stop* etc. Our system uses this hierarchical activity representation where activities can be decomposed down to atomic activities that are not further decomposable. The activity log is then a forest of trees representing high level activities, children of every activity are also activities ordered by time of their start. Each point in time has associated *activity trace* which is a trace from the high level activity down to the atomic activity.

We believe that this hierarchical activity representation will make the lifelog more accessible to a human user. Users will be able to “zoom” their activity to the level of detail that suits their needs, thereby focusing their search.

3 System architecture

The overall architecture of the system is shown in Figure 1.

1. The Android mobile phone logs GPS location and presence of WiFi and Bluetooth signals. During learning phase, user inputs activity annotation through a simple GUI. The logged data are stored in an internal SQLite database.
2. There is a set of desktop command line utilities used to extract features that are later used in the machine learning phase. For instance at this point the online geospatial database Gowalla [Gowalla, 2011] is used to add points of interest (POIs) to places detected in the movement log.
3. Machine learning is done inside the RapidMiner[1], an open source machine learning framework.
4. A GUI application is used for viewing and editing the data both from the database with the lifelog and for results of the machine learning. Figure 2 shows a screenshot of the GUI.

3.1 Implementation

Both the Android client and desktop applications were developed using Java. The desktop GUI log viewer and the editor was developed on the top of the Netbeans RCP[2]. Machine learning was performed mainly using RapidMiner. Since RapidMiner does not support Hidden Markov Model (HMM) used in some of our experiments, we used the JAHMM library[3] which implements HMM. We created a plugin[4] for RapidMiner that provides JAHMM's functionality. The plugin was released under a GNU GPL license.

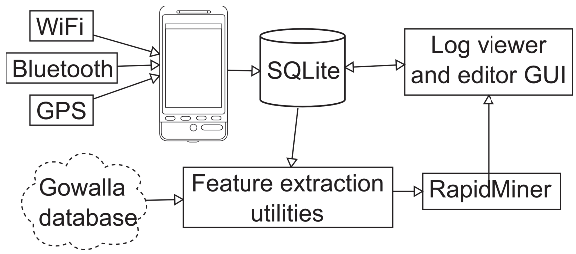

Figure 1: Architecture of the system.

4 Data collection

In this paper we present data collected by one participant between mid November 2010 and April 2011. We have other three participants who collected data over shorter time periods but each used different sets of activities. The application allowed users to mark beginnings and ends of their activities and use custom hierarchy for their description.

The GPS, WiFi and Bluetooth data was collected with period from 10 seconds to 2 minutes. Longer period was used to decrease power consumption when the phone was going out of power.

[1]RapidMiner 5.1 Homepage, `http://sourceforge.net/projects/rapidminer`, March, 2011.

[2]Netbeans RCP, Oracle Corporation, `http://platform.netbeans.org`, March, 2011.

[3]JAHMM Homepage, `http://http://code.google.com/p/jahmm`, March, 2011.

[4]The plugin is available for download at `http://http://code.google.com/p/rapidminerhmm/`, April, 2011.

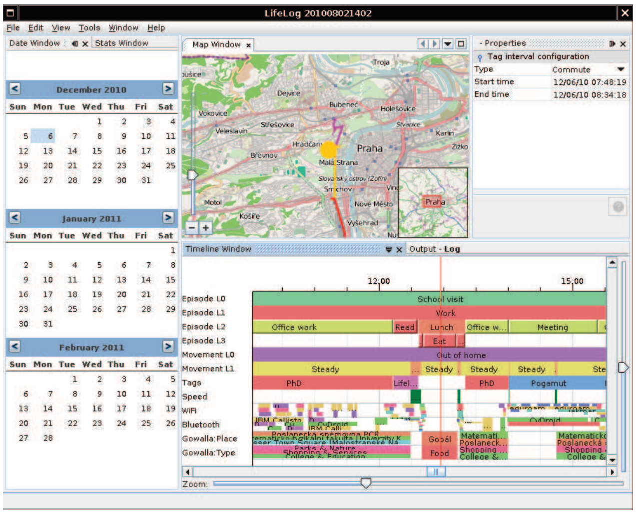

Figure 2: Screenshot of the desktop log editor. The editor contains a map view showed in the center, a timeline with activity log in the bottom and a calendar for time interval selection in the left.

5 Related work

In recent years several human activity recognition algorithms were published. They differ in sensors used as an input to the system (GPS/WiFi/accelerometer/video). Time scales considered in these recognizers vary from hours to weeks. As a formal model for activity recognition Dynamic Bayes Networks, including variants of Hidden Markov Models, are the most popular option. Table 1 shows related work published in the recent years.

Considering indoor activity recognition most of the systems use setup of accelerometers bound to specific parts of the body [Tapia, 2008; Huynh *et al.*, 2008; Stikic and Schiele, 2009]. This restriction is not applicable when considering normal mobile phone usage. Works using mobile phones without significant restriction on their usage [Lu *et al.*, 2010] predicted only classes like *walk*, *run*, *cycling* etc. [Lu *et al.*, 2010] presents system implemented in a mobile phone that is optimized for lower power consumption. [Liao *et al.*, 2007] is the closest to our work's aim. Iit features hierarchical activity representation but operates only in outdoor environments. An example of work from different domain is [Blaylock and Allen, 2006]. It performs activity recognition using hierarchy of Hidden Markov Models on data generated by a planning algorithm. Inputs of this system are purely symbolic, thus it allows for a higher level activity inference.

Citation	Hier. act.	Input data	Time scale	Environment	Algorithm
[Lu *et al.*, 2010]	×	GPS, Audio, Accel.	Days	City	SVM, GMM, NB, DT
[Liao *et al.*, 2007]	√	GPS	1month	City	DBN
[Huynh *et al.*, 2008]	√	Accelerometer	1 week	Indoor	SVM, HMM, NB, LDA
[Stikic and Schiele, 2009]	×	Accel.	Days	Indoor	Multi instance SVM
[Oliver and Horvitz, 2005]	√	A/V, Keyboard, Mouse	Hours	Indoor	DBN, HMM
[Yin *et al.*, 2004]	×	WiFi	Hours	Indoor	DBN, N-gram
[Tapia, 2008]	×	Accel., heart rate	Hours	Indoor	DT, NB
[Kautz *et al.*, 2003]	√	Noisy location	3 weeks	Indoor	Hierarchical HSMM
[Blaylock and Allen, 2006]	√	Artificial symbols	5000 plans	Monroe corpus	Hierarchical HMM

Table 1: Several existing activity recognition algorithms. Shortcuts used for algorithms: SVM = Support Vector Machine, GMM = Gaussian Mixture Model, NB = Naive Bayes, DT = Decision Tree, CRF = Conditional Random Field, LDA = Latent Dirichlet Analysis, DBN = Dynamic Bayes Network, HMM = Hidden Markov Model, HSMM = Hidden Semi-Markov Model.

6 Experiments

We have made two experiments with the logged data just to test the applicability of well known machine learning algorithms oo our data. In the first experiment we tested the ability to infer low level atomic activities like sleeping, working, hygiene etc. The second experiment focuses on inference of high level activities like visiting school, friends or parents, shopping, training etc.

6.1 Low level activities inference

Method

Logged data were transformed into feature vectors $f_1, f_2 ... f_T$. Each feature vector $f_t = \langle lat_t, lon_t, speed_t, hour\_of\_day_t, w_t^1 ... w_t^m, b_t^1 ... b_t^n, g_t^1 ... g_t^o \rangle$, where $w_t^i \in \langle 0, 100 \rangle$ is a WiFi network's signal strength $w^i \in W = \{$*a WiFi network whose first and last occurrence were at least 1 week apart and it was present for at least 4 hours it total*$\}$ in time t, $b_t^j \in \{0, 1\}$ indicates presence of a Bluetooth device $b^j \in B = \{$*a Bluetooth device whose first and last occurrence were at least 1 week apart and it was present for at least 30 minutes it total*$\}$, finally $g_t^k \in \{0, 1\}$ indicates presence of a place obtained from the Gowalla database with type $k \in \{$*Travel, Food, Parks & Nature, Shopping, Entertainment, Architecture & Buildings, College & Education, Nightlife, Art*$\}$ in time t. The feature vectors were sampled at a constant rate of 1 minute. There were 30 different types of atomic actions, the actions were: *Alpine skiing, Car repair, Clean car, Concert, Cook, Cross-country skiing, Cycling, Eat, Hair cut, Hand work, Home Office, Household, Hygiene, Idle, Meeting, Other, Packing, Play games, Program, Shop, Sleep, Spinning, Strengthening, Teaching, Travel, Wait, Walk, Watch TV, Working, Writing.*

In preliminary experiments we tested a CART decision tree [Breiman *et al.*, 1984], Hidden Markov Model [Rabiner, 1989], 1-NN classifier [Hart, 1967] and a zero classifier that predicts the most probable class no matter what the sensory input is. The decision tree, the zero classifier and the k-NN were used directly on a sequence of feature vectors. In case of the HMM feature vectors were clustered using k-means clustering into 1000 and 4000 clusters used as discrete observations. Hidden states were atomic actions, matrices of observation probabilities for states and state transitions were computed directly from the data. Laplace correction was used, hence none of the probabilities was zero. A Viterbi algorithm [Rabiner, 1989] was used for inference of the most probable sequence of hidden states.

Method	Accuracy in %
Zero classifier	32.3
1NN	48.5
HMM (1000)	50.0
Decision tree	50.9
HMM (4000)	52.1

Table 2: Comparison of performance of Zero classifier, Decision tree, 1-NN and two variants of HMM

Results

Table 2 shows performance of tested algorithms. The best performing was Hidden Markov Model (HMM) with observation space clustered into 4000 observations, but its accuracy was only 52.1%

The zero classifier predicted *Sleep* that was the most frequent class with almost 8 hours of sleep a day, this lead to accuracy of 32.3%. The other three classifiers performed comparably well with accuracy around 50%. The Hidden Markov Model succeeded in capturing several temporal dependencies in the data. For example most days begin with sequence: *sleep, hygiene, eat*, which was correctly revealed by the HMM.

The class best predicted by the HMM was *Sleep* with precision of 95% and recall of 89%, class *Work* had precision 73% and recall of 70%. Other classes were predicted with significantly lower accuracy.

Discussion

The performance of 52% is not satisfactory. This could be caused by presence of too many classes and by lack of some important information in the context provided to the machine learning. Based on this results and related works we have extended the logging application with a pedometer that will be used in future experiments. Sleep was predicted relatively well because this activity was bound with a specific place that was infered from WiFi network's signals. We originally thought that inclusion of Bluetooth data will increase recognition rate of activities like *Meeting* that can be bound to presence of specific people. However due to the fact that most people have switched off the Bluetooth discovery mode of their mobile phone this does not proved to be right.

6.2 High level activities inference

Method

In this experiment we wanted to automatically label high level activities. In the first step boundaries of high level activities were identified using the GPS data. Segmentation was performed by identifying intervals when the user was at home and when he was outside the home location. Algorithm 1 was used to identify these $outOfHome$ locations. Then for each interval corresponding to one $outOfHome$ log the activity logged by the user that overlapped it best was searched and the $outOfhome$ log was labeled with the name of this activity.

After this segmentation 113 distinct activities were found, 7 of these were assigned a unique label, these were removed because there will be no data left to split them into training and testing sets. The remaining 106 activities were used for the rest of the experiment, Table 3 shows distribution of classes in the data set.

For each $outOfHome$ entry a feature vector f was constructed. $f = \langle$*the length of the interval, the distance traveled, the time spend in movement/time without movement, the time of the day when the activity started, the time of the day when the activity ended, average speed when moving, the east, west, south and north most locations in that interval,* $w_1, ..., w_o, g_1, ..., g_p\rangle$, where w_i represented WiFi networks obtained as in the previous experiment, the same applies to the g_i Gowalla places.

Algorithm 1 Movement segmentation

Require: $locations$ — sequence of GPS locations
1: $filteredLocations \leftarrow$ all locations from $locations$ with accuracy better than 80 meters
2: $parts \leftarrow$ identify intervals of movement and intervals without movement from $filteredLocations$, remember the location of intervals without movement
3: $averageHomeLocation \leftarrow$ find a location that occurs most often at 3 a.m. of each day from the interval, this is considered to be the home location
4: find sequences of parts from $parts$ list that begin and end near the $averageHomeLocation$, add each such sequence to the $outOfHomeList$
5: **return** $outOfHomeList$ %% list with entries corresponding to intervals when the user was outside the home location

Results

Because our dataset was relatively small and some classes were represented by very few examples we used the leave one out cross validation. This means that we always build a classification model from $n-1$ examples and used it to predict the n-th example. Again as in the previous experimented we tested several machine learning algorithms. The best performing was the CART decision tree, the accuracy of classification was 67.92% (1-NN 42%, Naive bayes 51%). Table 4 shows confusion matrix of the best classifier.

Class name	Instances
School visit	23
Shopping	16
Training	15
Visiting friends	12
Work day	12
Weekend trip	11
Visiting parents	10
Trip	5
Visiting doctor	2

Table 3: Class distribution of the activities

Discussion

As can be seen the classifier performed relatively well. There is a mutual confusion between *School visit* and *Work day* classes because both involve traveling throuhg the same part of the city. The *Trip* and *Shopping* classes are classified relatively bad. This can be caused by high variance inside those classes, the shopping activity involved several shops and there were several different targets of trips. The data from Gowalla database did not help in this case, inspection of the learned decision trees showed that the Gowalla places were not used.

The main problem of this approach is the assumption that each $outOfHome$ segment corresponds to only one activity class. It is often the case that the segment consists of 8 hours of *School visit* followed by 30 minutes of *Shopping* and finally 2 hours of *Visiting parents*. In the current procedure this whole segment would be labeled as a *School visit*. Finer grained segmentation remains as future work.

7 General discussion and Future work

Accuracy of low level activity recognition has to be improved to match result reported in [Lu *et al.*, 2010] where mobile phones were also used to collect data. Higher level activity recognition provides better results and it is closer to use in real lifelogging applications. Future directions of work on our system include:

- Inclusion of accelerometer data — this should increase accuracy of low level action inference.
- Connection of low level and high level activities inference — high level activity recognition performed better than low level one. This could be used to create a two level recognizer where a high level activity can be used to restrict possible lower level activities thus improving the lower level classifier's performance.
- Finer grained activity segmentation — the procedure segmenting activities based on home location can be used to provide rough bounds that can be later refined by a different segmentation technique. We want to try HMM or Conditional Random Fields for this purpose.
- Inclusion of long term time dependencies — from the collected data we know that e.g. *Training* occurs twice a week whereas *Shopping* occurs usually once a week. Explanations of activity that are in accordance with this prior knowledge could be then preferred.

		True class									
		SV	VP	WT	VF	WD	TRI	TRA	SH	VD	precision
Predicted class	School visit (SV)	18	0	0	0	2	0	0	0	0	90%
	Vis. parents (VP)	0	6	0	0	0	0	0	4	0	60%
	Weekend trip (WT)	1	0	11	0	0	1	1	0	0	79%
	Vis. friends (VF)	1	0	0	8	0	0	1	1	0	73%
	Work day (WD)	3	0	0	1	7	0	0	3	0	50%
	Trip (TRI)	0	1	0	1	0	1	0	0	0	33%
	Training (TRA)	0	0	0	1	3	1	12	1	0	67%
	Shopping (SH)	0	3	0	1	0	2	1	7	0	50%
	Vis. doctor (VD)	0	0	0	0	0	0	0	0	2	100%
	recall	78%	60%	100%	67%	58%	20%	80%	44%	100%	

Table 4: Confusion matrix for the high level activity classification.

8 Conclusion

We have presented initial work on our activity recognition system built on Android mobile phones. Performance of high level activities that were segmented using GPS data is promising, however the lower level activities inference has to be improved. Besides technical issues there are also law issues regarding collecting of WiFi and Bluetooth signals. For example in Czech Republic where the data were collected it is legal to store data about presence of mobile phone's Bluetooth if the identity of a phone's owner cannot be revealed from this data. Phone owner's written permission is required otherwise.

Acknowledgments

This work was supported by the grants GA UK 21809 and GA CR 201/09/H057 and by the research project MSM0021620838 of the Ministry of Education of the Czech Republic.

References

[Bell and Gemmell, 2009] C.G. Bell and J. Gemmell. *Total recall: how the E-memory revolution will change everything*. Dutton, 2009.

[Blaylock and Allen, 2006] N. Blaylock and J. Allen. Fast hierarchical goal schema recognition. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 796. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. Wadsworth & Brooks. *Cole, Pacific Grove, California, USA*, 1984.

[Consolvo *et al.*, 2008] S. Consolvo, D.W. McDonald, T. Toscos, M.Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, et al. Activity sensing in the wild: a field trial of ubifit garden. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1797–1806. ACM, 2008.

[Gowalla, 2011] Gowalla. Gowalla homepage. `http://gowalla.com/`, 2011.

[Hart, 1967] P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[Huynh *et al.*, 2008] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 10–19. ACM, 2008.

[Kautz *et al.*, 2003] H. Kautz, O. Etzioni, D. Fox, D. Weld, and L. Shastri. Foundations of assisted cognition systems. *University of Washington, Computer Science Department, Technical Report, Tech. Rep*, 2003.

[Kröse *et al.*, 2008] B. Kröse, T. van Kasteren, C. Gibson, and T. van den Dool. Care: Context awareness in residences for elderly. In *International Conference of the International Society for Gerontechnology, Pisa, Tuscany, Italy*, pages 101–105, 2008.

[Liao *et al.*, 2007] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

[Lu *et al.*, 2010] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84. ACM, 2010.

[Oliver and Horvitz, 2005] N. Oliver and E. Horvitz. A comparison of hmms and dynamic bayesian networks for recognizing office activities. *User Modeling 2005*, pages 199–209, 2005.

[Rabiner, 1989] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Stikic and Schiele, 2009] M. Stikic and B. Schiele. Activity recognition from sparsely labeled data using multi-instance learning. *Location and Context Awareness*, pages 156–173, 2009.

[Tapia, 2008] M. Tapia. *Using machine learning for real-time activity recognition and estimation of energy expenditure*. PhD thesis, Massachusetts Institute of Technology, 2008.

[Vemuri *et al.*, 2006] S. Vemuri, C. Schmandt, and W. Bender. iRemember: a personal, long-term memory prosthesis. In *Proceedings of the 3rd ACM workshop on Continuous archival and retrival of personal experences*, pages 65–74. ACM, 2006.

[Yin *et al.*, 2004] J. Yin, X. Chai, and Q. Yang. High-level goal recognition in a wireless LAN. In *Proceedings of the national conference on artificial intelligence*, pages 578–584. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[Zacks and Swallow, 2007] J.M. Zacks and K.M. Swallow. Event segmentation. *Current Directions in Psychological Science*, 16(2):80, 2007.