How Busy is my Supervisor?

Detecting the visits in the office of my supervisor using a sensor network

Ahmed Nait Aicha
Department of Computer Science
HvA University of Applied Sciences
Amsterdam, The Netherlands
a.nait.aicha@hva.nl

Gwenn Englebienne
Department of Computer Science
University of Amsterdam
Amsterdam, The Netherlands
g.englebienne@uva.nl

Ben Kröse
HvA University of Applied Sciences &
University of Amsterdam
Amsterdam, The Netherlands
b.j.a.krose@hva.nl

ABSTRACT

Existing research on the recognition of Activities of Daily Living (ADL) from simple sensor networks assumes that only a single person is present in the home. In real life there will be situations where the inhabitant receives visits from family members or professional health care givers. In such cases activity recognition is unreliable. In this paper, we investigate the problem of detecting multiple persons in an environment equipped with a sensor network consisting of binary sensors. We conduct a real-life experiment for detection of visits in the office of the supervisor where the office is equipped with a video camera to record the ground truth. We collected data during two months and used two models, a Naive Bayes Classifier and a Hidden Markov Model for a visitor detection. An evaluation of these two models shows that we achieve an accuracy of 83% with the NBC and an accuracy of 92% with a HMM, respectively.

Categories and Subject Descriptors

I.5.1 [**PATTERN RECOGNITION**]: Models—*Statistical*; I.5.4 [**PATTERN RECOGNITION**]: Applications—*Signal processing*; H.1.2 [**MODELS AND PRINCIPLES**]: User/Machine Systems—*Human information processing*; I.2.1 [**ARTIFICIAL INTELLIGENCE**]: Applications and Expert Systems—*Office automation*; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*Parameter learning*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Ambient assisted living and health monitoring, Sensor networks for pervasive health care, Naive Bayes Classifier, Hidden Markov Models

PETRA'12, June 6 - 8, 2012, Crete Island, Greece.

1. INTRODUCTION

A large number of countries face severe population ageing in the near future. As a consequence of this, the cost of health care is expected to grow enormously the coming years. One way to keep these costs limited is to introduce assistive technology that offers physical help, cognitive help or social help. For this, sensing systems are needed that monitor the patient. Networks of simple sensors are becoming popular for measuring activities. This can be simple actions, like in fall detection systems, or more complex activities. A large group of monitoring systems focuses on recognizing ADL: activities performed on a daily basis such as sleeping, toileting and cooking. The list of ADL was set up by [8], and is used by professional caregivers for monitoring the wellbeing of a person, in particular of elderly, by registering how well ADL are performed over time. In automatic recognition systems very often probabilistic models such as the hidden Markov model (HMM) [14] or conditional random fields (CRF) [13] are used to map the observed sensor data onto the hidden activity states. However, all presented research on the recognition of ADL from simple sensors assumes that only a single person is present in the home.

In our group we do long term experiments where we try to recognize ADL from sensor networks. We monitor a number of elderly for more than 6 months, and study trends and correlations with ADL measurements from professional nurses. For that reason it is important to be sure that the data we are collecting indeed originates from a single person, and to discard data of that is caused by multiple persons in the home. In this paper we present our work on detecting the number of people in an environment equipped with a simple sensor network. Given the application, we simplify the problem to detecting whether there is a single person or multiple persons in the environment. As a test situation we equipped the office of the supervisor with multiple sensors.

In section 2 we describe related work. In section 3 we describe our approach, where we compare a naive Bayes classifier with a HMM approach. The experiments and set-up are described in section 4. Section 5 describes the results and section 6 concludes the paper.

2. RELATED WORK

Much research has been carried out on detecting, counting and tracking multiple people and monitoring their activities using video cameras [5]. Real time visual surveillance systems for outdoor environment already exist [6, 7]. However, there is much less research on the detection and tracking of multiple people with sensor networks, especially those with simple binary sensors. A lot of research has focused on recognizing the ADLs of a single person. Different kinds of methods like Hidden Markov Models (HMM) [14], hierarchical hidden Markov model (HHMM) [10], Switching hidden semi-Markov models (SHSMM) [2], Rao-Blackwellised particle filter [12] have been used for modelling and inference of the ADLs of a single person.

In [16] the problem of simultaneous tracking and activity recognition (STAR) was introduced. The goal of STAR is to track multiple persons in a home setting and to recognize their activities. From an experimental setting where mainly motion sensors and contact switches are installed, it is shown that the use of Particle Filtering method has potential to solve the STAR problem. However, the accuracy of the tracking decreases as more people are in the house. In [1] two HMM's (an activity model and a person model) have been used to detect a human social interaction in an office. Unfortunately a deeper insight into the methodology and accuracy is not given in the paper. In [4], [15] and [3] an emerging pattern based multi-user recognizer (epMAR) to recognize both single-user and multi-user activities is proposed. The sensor platform used for this experiment mainly consists of wearable sensors. The sensors are not only simple binary sensors, but also sophisticated sensors like acceleration, location, sound and voice sensors. The focus of this paper is detecting the presence of one or more persons in an environment using a simple sensor network consisting of binary sensors.

3. APPROACH

The sensor network we use to count the number of persons in a room consists of motion detectors and a switch on the door, giving a binary output vector $O^{(t)}$ at every time stamp t when one of the sensors changes value. We define

- The set $S = \{s_1, ..., s_{|S|}\}$ as the set of sensors installed in the office of the supervisor, where $|S|$ is the total number of sensors.
- The observation $O^{(t)} = (o_1^{(t)}, \dots, o_{|S|}^{(t)})$ as the vector of sensor events at time stamp t generated by the $|S|$ sensors. $o_i^{(t)} = 1$ if the i-th sensor generates an event at time t, otherwise $o_i^{(t)} = 0$. For notation simplicity, the superscript (t) will be discarded.

In this paper we want to classify whether there is a single person or multiple persons in the room and compare two classifiers. The first classifier is a Naive Bayes Classifier, that for every event at time stamp t gives a class on the basis of the observation $O^{(t)}$. The second classifier is a Hidden Markov Model, that classifies the entire observation sequence $O^{(1)}, \dots, O^{(T)}$. We first describe the method we used for feature selection and then describe the classifiers.

3.1 Feature selection

The process of feature selection we have used consist of defining a set of basic features and a set of logical operations on these basic features. In our experiments we then generate all combinations of these basic features and the logical operators, and test which feature set performs best. To this end we have selected for every sensor event o_i at time stamp t two basic features $x_i = o_i$ and $x_{i+|S|} = 1 - o_i$. On every possible combinations of these $2|S|$ features, which are in total $\sum_{i=1}^{2|S|}(2|S|)!/((2|S|-i)!i!)$ combinations, we applied the methods Naive Bayes Classifier and the Hidden Markov Models.

3.2 Method: Naive Bayes Classifier

The Naive Bayes Classifier is used as a visitor detector. Therefore we define:

- The set $C = \{1, 2\}$ as the set of classes. $c = 1$ if the number of persons in the office is exactly one, i.e. there is no visit. $c = 2$ if the number of persons in the office is more than one i.e. there is a visit. For notation simplicity, C will be used in the equations i.s.o. $|C|$ to denote the total number of classes.
- The vector of classes as Y. $Y_t = 1$ if there is exactly one person (usually the supervisor) in the office at time stamp t. $Y_t = 2$ if there are two or more persons in the office.
- The vector $X = (x_1, x_2, ..., x_D)$ as the feature vector of an observation $o^{(t)}$. The feature vector is binary, so $x_i = 1$ if the i-th feature is true, otherwise $x_i = 0$.

To detect the visits of the supervisor at the office at some time stamp t, We are interested in calculating the posterior probability defined by $P(Y|x)$. to calculate this probability we apply the Bayes rule, as given in equation 1.

$$\begin{aligned} P(Y|X) &= \frac{P(X|Y)P(Y)}{P(X)} \\ &= \frac{P(X|Y)P(Y)}{\sum_{y=1}^{C} P(X|y)P(y)} \end{aligned} \tag{1}$$

For each class c, the class prior π_c is defined in equation 2.

$$\pi_c := P(Y = c) \tag{2}$$

As the feature vector is binary, We may calculate the MLE $\hat{\pi}$ of π as given in equation 3.

$$\hat{\pi} = (\pi_1, \dots, \pi_C) = (N_1/N, \dots, N_C/N) \tag{3}$$

where N_c is the number of training examples that have class label c and N the total number of the training examples.

Assuming X has dimension D and the features are i.i.d, we can use a multinomial distribuion for the conditional probability P(X|Y). Denoting Θ, the parameters of this multinomial distribution, then the conditional density function given a class c is given in equation 4.

$$P(X|Y = c; \Theta) = \prod_{i=1}^{D} P(x_i|Y = c) \tag{4}$$

where,

$$P(x_i|y = c; \theta_c) = \theta_c^{x_i}(1 - \theta_c)^{1-x_i} \tag{5}$$

The MLE $\hat{\Theta}$ of Θ is given in equation 6.

$$\widehat{\theta_{ijc}} = \frac{\#\{X_i = x_{ij}, Y = c\}}{\#\{Y = c\}} \tag{6}$$

To avoid the problem of zero counts which will result in zero probabilities in the equation 4, we use La Place smothing estimator for Θ given in equation 7.

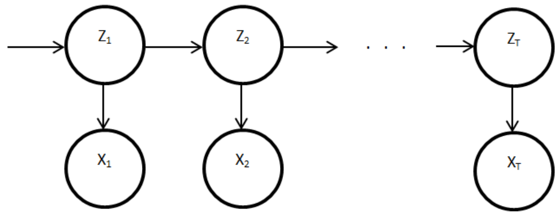

Figure 1: Graphical view of a first order Hidden Markov Model

$$\theta_{ijc}^{laplace} = \frac{\#\{X_i = xij, Y = c\} + 1}{\#\{Y = c\} + C} \quad (7)$$

Given a test feature x which is extracted from an observation o, we can calculate the posterior probability $P(Y = c|x)$ for each class c by using the Bayes rule given in 1 and the estimates given in 3 and 7. If the posterior probability $P(Y = c|x)$ is bigger than 0.5 then the predicted class is c, as we only have two classes.

For the implementation of the Naive Bayes Classifier, the class NaiveBayes of the numerical computing environment Matlab has been used.

3.3 Method 2: Hidden Markov Models

The problem of detecting visits can also be tackled using Hidden Markov Models (HMM). A graphical representation of the first order HMM is given in Figure 1. In this Figure, the observation vector at time t denoted by x_t is represented by the features selected in section 3.1. The hidden state at time t denoted by z_t is represented by the visits to the supervisor at the office. $z_t = 1$ if the number of persons in the office is exactly one i.e. there is no visit. $z_t = 2$ if the number of persons in the office is two or more i.e. the is a visit. The HMM is mathematically represented by 8.

$$\lambda = (A, B, \pi) \quad (8)$$

where $A = \{a_{ij} : 1 \leq i, j \leq N\}$ defines the transition probability matrix, $B = \{b_j(k) : 1 \leq j \leq N, 1 \leq k \leq M\}$ defines the emission probability matrix and $\pi = \{\pi_i : 1 \leq i \leq N\}$ defines the initial state probability. a_{ij} is the transition probability of taking the transition from state s_i to state s_j. $b_j(k)$ is the conditional probability of emitting the $k - th$ observation given the state s_j. The set of the states, given by $S = \{s_i : 1 \leq i \leq N\}$, is represented by the number of persons (one or more than one) in the office. The set of the observations, given by $V = \{v_k : 1 \leq k \leq M\}$, is represented by the feature space discussed in section 3.1. The dimension of V is equal to N^d, where d is the number of features used.

The parameters (A, B, π) of the HMM used in equation 8 are estimated using the maximum likelihood estimates.

Given a sequence of features $x_{1:T}$ with length T, we want to predict whether the supervisor has a visit at the office during this period of time. For this reason we calculate for each time stamp t $(1 \leq t \leq T)$ the posterior state probability. This posterior is, for the visits state, defined as the conditional probability of being at state s_2 at time stamp t, given the observed sequence of features $x_{1:T}$. Calculating this posterior probability for every t $(1 \leq t \leq T)$, will result in a vector of posterior state probabilities denoted by $p(z_t = s_2|x_{1:T})$. Counting the number of posteriors with values bigger than 0.5, the probability of a visit is then equal to number of this counts divided by the length of the posteriors vector, which is equal to T. The general posterior state probability is given in equation 9. In the first equality we applied the Bayes rule and in the second equality, we write the vector $x_{1:T}$ as $(x_{1:t}, x_{t+1}, x_{t+1:T})$ and apply the conditional independence property together with the product rule of probability.

$$\begin{aligned} p(z_t|x_{1:T}) &= \frac{p(x_{1:T}|z_t)p(z_t)}{p(x_{1:T})} && (9) \\ &= \frac{\alpha(z_t)\beta(z_t)}{p(x_{1:T})} && (10) \end{aligned}$$

where

$$\alpha_i(z_t) := p(x_{1:t}, z_t) \quad (11)$$

and

$$\beta_i(z_t) := p(x_{t+1:T}|z_t) \quad (12)$$

$\alpha_i(z_t)$ represents the joint probability of all the observations up through time t and that we are in state s_i at time t and $\beta_i(z_t)$ represents the conditional probability of the future observations from time $t + 1$ up to time T given the state state s_i at time t.

The forward probabilities $\alpha_i(z_t)$ and the backward probabilities $\beta_i(z_t)$ can be iteratively computed using dynamic programming algorithms called the Forward and Backward Procedure.

For the implementation of the estimation of the parameters and the inference we used the HMM functions hmmestimate and hmmdecode of the numerical computing environment Matlab.

4. EXPERIMENTAL SET-UP

Our test-bed consists of the office of the supervisor as shown in Figures 2 and 3. In the office a sensor network as described in [14] and consisting only of binary sensors, has been installed. The type of the sensors and their place in the office are listed in Table 4. We recorded the visits to the office during two months with a video camera installed in a corner of the office. To produce accurate and easy to make annotations we stored the video images of the camera every one minute to a different file. The days that the supervisor is not in the office or the video camera was not able to store images are removed from the data set. The sensor events we used are stored in seconds precision while the annotation is made in minutes precision. This means that when the supervisor has a visit at some time stamp $hh : mm : ss$, all the sensor events within the same minute, including the events just before the beginning of the visit, are annotated as a visit. Using the exact annotation, we should annotate all the events with a second part smaller than ss as having no visit and all the events with a second part bigger or equal than ss as having a visit. To deal with this discrepancy in precision, we conducted an experiment to define the optimal length of the sensor event to use. This experiment and the corresponding results are described in 5.2.

The total number of annotated sensor events recorded this two months was 26543, where a percentage of 45% belongs

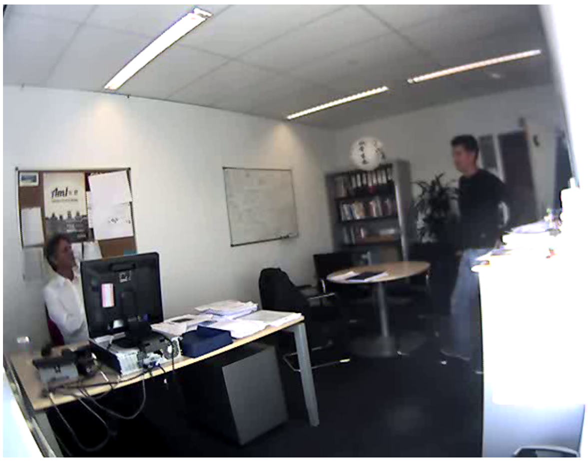

Figure 2: The office of the supervisor at the research centre: the supervisor sitting on the chair at his desk having a short visit of a researcher.

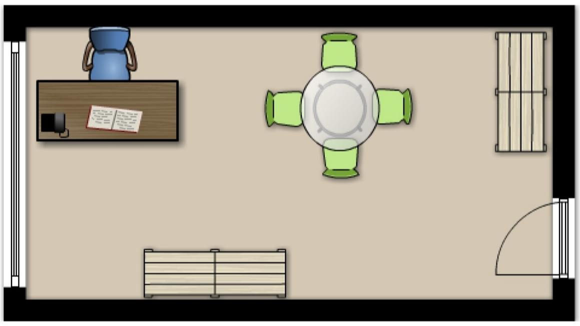

Figure 3: The top view layout of the office showing the places of the sensors described in Table 4

to the visits class and 55% tot the no visits class. An example of sensor events during a day is shown in Figure 4. The corresponding annotated number of persons in the office during the same day is shown In Figure 4. From these two figures we can see that when the supervisor is alone sitting at his desk (between 09:41 and 09:59 and between 17:17 and 18:38), the sensor with id equal to 5 generates the most events. When the supervisor has a meeting at the round table, the sensor with id 3 generates the most events. In case the number of persons participating to the meeting is more than 3, the sensor with id 4 also generates events. Note that even there is no person in the office some sensors may generate some noisy event like sensor with id 5 at 11:57 and 15:08.

5. RESULTS

5.1 imbalanced class sets

A confusion matrix as shown in Table 5.1 is typically used to evaluate performance of a machine learning algorithm. The overall classification accuracy, based on the confusion matrix and defined by $Acc = (TP + TN)/N$ is very often used to measure the performance of the classifier. In the conducted experiments (especially when testing the per-

Sensor id	Sensor type	Place in the office
1	switch	on the door
2	movement	above the door
3	movement	under the meeting table pointing to the passage
4	movement	under the meeting table pointing to the wall
5	movement	above the desk

Table 1: The type and place of the sensors installed in the office. See also Figure 3

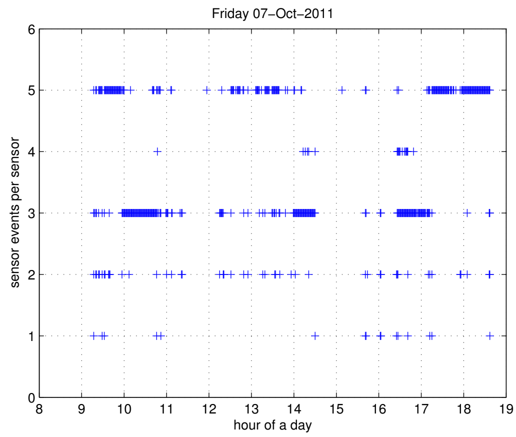

Figure 4: The sensor events during a working day. The type and location of each sensor is given in Table 4

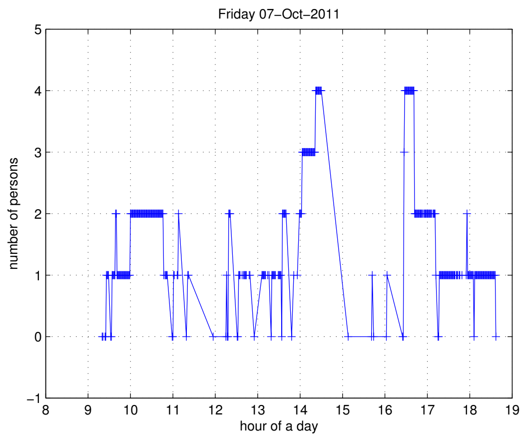

Figure 5: The number of persons in the office during a working day.

formance of HMM with a relative big sequence length) the percentage of the sequences belonging to class visits may be too small comparing to the percentage of the 'no visits' class. Using the traditional overall accuracy for this case is therefore inappropriate. The reason is that when applying a naive classifier which assigns all the complete test data the majority class will lead to a high accuracy while the minority class is totally misclassified. One way to deal with this imbalance problem is by under-sizing the majority class or over-sizing the minority class. However, The disadvantage is that over-sizing will lead to loss of information while under-sizing may lead to overfitting [11]. Another way to deal with this problem is to assign a high cost to the misclassification of the minority class and a low cost to the misclassification of the majority class while trying to maximize the overall accuracy. In other words, it is desirable to have a metric that gives high prediction accuracy over the minority class, while maintaining reasonable accuracy for the majority class. To this end we used the Geometric Mean [9] given in equation 13 which maximizes the accuracy of each class while keeping these accuracies balanced. Note that Acc^+ is the same as the recall of the positive class and Acc^- the recall of the negative class. Acc^+ is also called the true positive rate and Acc^- is called the true negative rate.

		Predicted class		Total
		yes	no	
Actual class	positive	TP	FN	
	negative	FP	TN	
	Total			N

Table 2: Confusion Matrix

$$Acc_{gm} = \sqrt{Acc^+ \times Acc^-} \quad (13)$$

$$= \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \quad (14)$$

5.2 Naive Bayes Classifier

In order to be able to apply the NBC as described in section 3.2, the raw data is needed to be processed. To this end, we first removed the sensor data which is not annotated. Then we removed the data corresponding to the absence of persons in the office. The remaining data, denoted as the complete data set, is split into two categories:

- The category training set containing 60% of the complete data set. This set is used to estimate the parameters of the NBC.
- The category test set containing 40% of the complete data set. This set is used to determine the accuracy of the NBC.

The sensor events we used are stored in seconds precision while the annotation is made in minutes precision. Using one minute sample resolution will lead to loss of data while using one second sample resolution may affect the speed of the classifier. We conducted therefore different experiments to define the optimal sample resolution. In these experiments we use eight different sample resolutions $r \in \{1, 2, 5, 10, 15, 20, 30, 60\}$ for the time stamp of the sensor event. Using a sample resolution r means that the seconds

event time stamp	$r = 5$	$r = 10$	$r = 15$
hh:mm:04	hh:mm:00	hh:mm:00	hh:mm:00
hh:mm:05	hh:mm:05	hh:mm:00	hh:mm:00
hh:mm:07	hh:mm:05	hh:mm:00	hh:mm:00
hh:mm:10	hh:mm:10	hh:mm:10	hh:mm:00
hh:mm:22	hh:mm:20	hh:mm:20	hh:mm:15
hh:mm:45	hh:mm:45	hh:mm:40	hh:mm:45
hh:mm:59	hh:mm:55	hh:mm:50	hh:mm:45

Table 3: A set of examples of event time stamps and the corresponding rounded time stamps when using a sample resolution $r \in \{5, 10, 15\}$. For simplification of the notation the date part of the event is discarded.

sample resolution (sec)	overall accuracy (%)	geometric mean (%)	best feature combination
01	87	83	[3,5]
02	87	83	[3,5]
05	86	83	[3,5]
10	86	83	[1,2,3,8],[2,3,5,10]
15	85	82	[1,2,3,8]
20	85	82	[1,2,3,8]
30	85	81	[2,3,4,5,10],[1,2,3,8]
60	84	80	[2,3,4,5,10],[1,2,3,8]

Table 4: The overall accuracy, the Geometric mean and its best feature combinations corresponding to different sample resolutions in seconds. The values are the averages over 30 random experiments.

part s of a time stamp of a sensor event is rounded according to the function $f_r(s) = r.(s/r)$, where '/' denotes the integer division. Table 5.2 gives a set of event time stamps and the corresponding rounded time stamps when using a sample resolution $r \in \{5, 10, 15\}$. Note that when using resolution $r = 1$ the event time stamps remain the same and when using the resolution $r = 60$ the seconds parts of the time stamp are set to 00.

The experiments conducted are repeated 30 times. For theses experiments the mean of the overall accuracy values, the mean of the Geomteric mean values and the corresponding best feature are calculated. The results of these experiments are given in table 5.2. The best feature combination is defined as the most (at least 50%) occurred combination(s). From this table we see that when using a small sample resolution ($r \in \{1, 2, 5\}$), only two features are needed to fit the data well and when using a bigger sample resolution ($r \in \{30, 60\}$) five features are needed to fit the data well. The results in Table 5.2, show also that the values of the overall accuracy and the geometric mean seem to decrease with the sample resolution, but not in a significant way.

5.3 Hidden Markov Model

In order to be able to apply the HMM as described in section 3.3, we have preprocessed the data on the same way as done for the Naive Bayes Classifier. The only difference is that for the NBC a set of single feature vectors for training and testing while for the HMM a set of consecutive feature

sample resolution (sec)	length sequence	overall accuracy(%)	geometric mean(%)	best feature combination
01	300	95	90	[3,7]
02	150	95	91	[3,7]
05	60	95	91	[3,5]
10	30	94	92	[3,4,6,7]
15	20	93	92	[3,7]
20	15	93	92	[3,5]
30	10	93	92	[2,3,5,8]
60	05	92	92	[2,4,7,9]

Table 5: Different sample resolutions (in seconds) and the corresponding length of the sequence (number of sensor events) of a time window of 5 minutes. For each each sample resolution the average overall accuracy, the average Geometric mean and its best feature combinations are calculated.

vectors $x_{1:N}$ with length N is used.

The length N of a sequence $x_{1:N}$ is dependent of the sample resolution we used. The values used are given in table 5.3. Note all the values correspond to a time window of 5 minutes. Hence, when using a sample resolution of 60 seconds, the length of a sequence is at most 5 and when using a sample resolution of 15 seconds, the length of the sequence is a most 20.

For each training set a random test set has been used to calculate the posterior state probabilities and therefore to determine if the sequence was well-classified or misclassified. A test sequence $x_{1:N}$ is well-classified if the majority (>50%) of the predicted states of this sequence is well classified.

To determine the performance of the HMM, we conducted an experiment on the same way as done for the NBC. The results of this experiment, as given in Table 5.3, show that the values of the overall accuracy (respectively the geometric mean) seem to decrease (respectively increase) with the sample resolution, but not in a significant way. Note that the best feature combinations given in Table 5.3 are one of the lot of combinations that fit the HMM well.

5.4 Comparison of the methods

The difference between the Naive Bayes Classifier and the Hidden Markov Model is that with the NBC the feature vector x_t consists of one observation at some time stamp t while the feature vector $x_{1:N}$ used for HMM consists of a sequence of observations with some length N. Comparing the results of the NBC as shown in Table 5.2 to the results of the HMM as shown in Table 5.3, we see that the HMM over-performs the NBC. This holds for both the overall accuracy as for the Geometric mean accuracy. These tables also show that the influence of the different sample resolutions on the performance of the models is not significant as the overall accuracy and the Geometric mean do not significantly increase or decrease with the sample resolution.

6. CONCLUSIONS

In this paper we studied the problem of detecting the presence of multiple persons in an environment. We conducted a real-life experiment where the office of the supervisor, equipped with a sensor network using binary sensors, has been monitored during two months. We applied two classification methods, the Naive Bayes Classifier and the Hidden Markov models.The results show the effectiveness of the HMMs.

As our group is interested in recognizing ADLs of elderly people, there are some limitations in this work. Although the used data set is real life, the experimental setting is limited to one room, the office, and the number of sensors is small. Also the type and the duration of the activities of the supervisor are different from these of elderly people. For a future work, we will apply (at least) the same techniques on a bigger sensor data set generated by a number of independently living elderly people we are monitoring at this moment. Their houses are equipped with the same sensor network, but have more than one room and the number of the sensors are much more than used in the office of the supervisor.

7. REFERENCES

[1] D. J. Cook, A. Crandall, G. Singla, and B. Thomas. Detection of social interaction in smart spaces. *Cybernetics and Systems*, 41(2):90–104, 2011/10/26 2010.

[2] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838 – 845 vol. 1, june 2005.

[3] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1 –9, march 2009.

[4] T. Gu, Z. Wu, L. Wang, X. Tao, and J. Lu. Mining emerging patterns for recognizing activities of multiple users in pervasive computing. In *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, pages 1 –2, july 2009.

[5] I. Haritaoglu, D. Harwood, and L. Davis. A real-time system for detecting and tracking people in ... In H. Burkhardt and B. Neumann, editors, *Computer Vision — ECCV'98*, volume 1406 of *Lecture Notes in Computer Science*, pages 877–892. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055710.

[6] I. Haritaoglu, D. Harwood, and L. Davis. Hydra: multiple people detection and tracking using silhouettes. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 280 –285, 1999.

[7] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809 –830, aug 2000.

[8] S. Katz. Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. *J. Am. Geriatrics Soc.*, 31(12):721–726, 1983.

[9] M. Kubat, R. Holte, and S. Matwin. Learning when negative examples abound. In M. van Someren and G. Widmer, editors, *Machine Learning: ECML-97*,

volume 1224 of *Lecture Notes in Computer Science*, pages 146–153. Springer Berlin / Heidelberg, 1997. note.

[10] S. Luhr, H. Bui, S. Venkatesh, and G. West. Recognition of human activity through hierarchical stochastic learning. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 416 – 422, march 2003.

[11] D. Mease, A. J. Wyner, and A. Buja. Boosted classification trees and class probability/quantile estimation. *J. Mach. Learn. Res.*, 8:409–439, May 2007.

[12] N. Nguyen, D. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 955 – 960 vol. 2, june 2005.

[13] T. T. Truyen, D. Phung, H. Bui, and S. Venkatesh. Hierarchical semi-markov conditional random fields for recursive sequential data. *In Neural Information Processing Systems (NIPS)*, 2008.

[14] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9, New York, NY, USA, 2008. ACM.

[15] L. Wang, T. Gu, X. Tao, and J. Lu. Sensor-based human activity recognition in a multi-user scenario. In M. Tscheligi, B. de Ruyter, P. Markopoulus, R. Wichert, T. Mirlacher, A. Meschterjakov, and W. Reitberger, editors, *Ambient Intelligence*, volume 5859 of *Lecture Notes in Computer Science*, pages 78–87. Springer Berlin / Heidelberg, 2009.

[16] D. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In H. Gellersen, R. Want, and A. Schmidt, editors, *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 329–334. Springer Berlin Heidelberg, 2005.