

Adaptive and Automated Ambiance Surveillance and Event Detection for Ambient Assisted Living

Dierck Matern, Alexandru Condurache, and Alfred Mertins
Institute for Signal Processing, University of Luebeck
Ratzeburger Allee 160, 23562 Lübeck

Abstract—In this paper, we discuss an event detection system using a wireless sensor network in the Ambient Assisted Living context. The sensors measure the environment in which the patients live, not vital parameters of the patient him- or herself, which is important in order to respect the privacy and informational self-determination of the patient. The major difficulties of the given setup with sensors in the environment are that the measurements are taken in an irregular fashion (as opposed to regular sampling) and that some of the sensors may be unreliable. To tackle these problems, we propose an event-detection framework that is based on the theory of conditional random fields [1]. We conduct experiments on real-life sensor data collected from a set of eight households. The experiments show that the conditional random field is well suited for ambiance surveillance.

I. INTRODUCTION

In Ambient Assisted Living (AAL), one important aim is to gather information on the condition of the resident and to detect when the behavior and/or health status changes. Only when the resident's condition is known, we can react to pathologic changes or emergencies, for example, by automatically sending a status message to a family member or calling the ambulance. However, a direct observation of the patient (i.e., the person who resides in an AAL supported residence), is not always suitable, because it may conflict with the acceptance of the patient and usually inflicts the patients privacy. A good AAL system thus has to respect the privacy and self-determination of the resident, but also has to provide the capability to detect whether an interference is necessary.

In this work, we limit the observation process to taking measurements within the residence and therefore we only measure the interaction of the resident with his surroundings, not the resident him- or herself. That is, we use sensors that neither have to be worn (for example an emergency watch) nor have to include direct interactions of the patient with the system (for example a button that has to be pressed regularly), nor interfere with the privacy of the patient, like cameras or microphones. This is an important difference to several other AAL systems, for example [2] In the paper, we discuss the statistical analysis of the data and describe a system for the detection of emergencies and changes in the resident's behavior without disclosing much information about the actual habits or activities and thus guaranteeing informational self-determination.

The situations which require a third party to react, for

example a medic or family member, are called *events*. Because these events are arbitrary, rare, and because we need to react the very first time they occur, it is not possible to train a classifier that makes a decision between several well stated and predefined situations. Therefore we need to model and understand the usual situations, i.e., the *normal case*, and then define the events as the absence of the normal case. This task will be called *event detection* in the remainder of the paper.

The proposed algorithm is based on the theory of conditional random fields [1], which are powerful abstract models for statistical inference. We describe this method in its basics. As an addition, we discuss two different strategies of preprocessing and compare them to decide which provides better results, and hence with which we proceed in further analysis. The method provides the possibility to be adapted to more specialized problems and additional tasks, as well as to additional sensors that may be installed at a later stage. Hence, the methods for the statistical analysis also have to be adaptive and scalable. For example, adaptivity is required, because the situations and furniture may change within a household in a normal manner. The algorithms to analyze the data need to have the possibility to deal with the removal of sensors while being able to process the data of the remaining sensors, but also may have to include new sensors. As mentioned above, we solve these problems by using Conditional Random Fields (CRFs) [3], [1], [4]. In CRFs, we can increase and decrease the number of sensors without retraining the whole model, as will be shown in this paper. Since CRFs do not depend on a certain (fixed) sampling rate, we can even use CRFs on different time scales. To discuss this adaptive setting is the main topic of this paper.

Since AAL monitoring systems are meant to be used over extended periods of time, we target to detect events on different time scales, from slow changes that can only be observed if we consider large periods, for example several months or years, to sudden events that can be detected if we consider only a few minutes. In the experiments of this paper, we discuss the latter case. Because the real-life testing data does not provide any events so far, and in order to have a ground truth for the experiments, events had to be artificially introduced by data manipulation. This will be described in more detail in the corresponding section.

The rest of the paper is structured as follows. In Section II, we first describe the currently employed sensors and the

feature extraction process. Then we discuss the theory of CRFs in the context of the given application. In Section III, we apply training and inference on real measured data and show that the CRFs can be used to model the data. In Section IV, we discuss our results and give an outlook of the provided methods.

II. THEORETICAL BACKGROUND AND INFERENCE

To understand the possibilities of CRFs in the context of event detection in the AAL setup, we first discuss the currently employed sensor network before we turn to the CRFs themselves.

A. Sensor Setup

The sensors are wireless devices with independent power supply. Each sensor sends its measurement to a central server, together with a time stamp and the battery status. The sensors are as follows: a door sensor which recognizes the movement of a specified door; a Passive Infra-Red (PIR) and motion sensor placed at the entrance to detect movement at the entrance door; a motion sensor placed at a specific door within the household; a PIR sensor to detect movement within a specific area; a motion sensor which measures the motion of furniture, usually placed at the bed; a sensor which measures temperature and humidity at the drainage. Not all households taking part in the initial studies have all sensors installed, and not all sensors are online all the time, so in the data, there is missing information, and the measurements for each sensor differ between the households because of different placements or floor plans.

The different kinds of sensors produce very different measurements. The door sensor, for example, sends "true" if the door was used and "false" otherwise, while the furniture sensor integrates the movements over a short period and therefore is a continuous measure. Hence, to apply a unified distribution to model the data and use tests like a likelihood ratio test is difficult, especially if we consider to have new kinds of sensors in the future. A generative model like a hidden Markov model [5] is also not suitable, because we either had to assume the distribution of the data, too, or we would have to estimate it, which can be difficult for unknown sensors. To overcome this problems, we use CRFs which are data-driven models [6]. CRFs use the data as such, without assuming independences, correlations or distributions among the data. We extend the CRF-based method proposed in [4] to multi dimensional, irregular sampled data.

Some sensors are inactive when the observed variables do not change, while others send the information frequently. Hence we need a strategy to gain data for each evaluation of the CRF. For this, we integrate the measured data for a certain time frame.

B. Conditional Random Fields

CRFs are data-driven discriminative, finite state models [1], [7]. They offer the possibility to conduct inference without the need to model the distribution of the data in a direct

manner. They transform the feature space to a time and state discrete space in which the Markov properties hold. It can be shown [1], [3] that the formulation of the CRF fulfills the statute of the maximum entropy. That is, depending on the feature extraction process, the CRFs have the potential to provide the best classifier for our problem.

In the following, we first discuss the abstraction of the feature vectors to the state space. Then we turn to the interpretation of the states and show how we can use them for the event detection. Therefore, we model the probability of a state sequence, given the measurements, and we use the likelihood of each state to decide if we have observed an event.

Let \mathbf{X} be a set of measurements. From this data, we have a vector representation $\phi(\mathbf{X}, n)$ at each time step n . These vectors represent either integrated time windows or interpolated values over time to get one vector at each time step, as discussed in Section II-A. We describe the vector representation in Section II-C.

A CRF determines a probability of a state sequence \mathbf{S} , given the sequence \mathbf{X} and a trained model [1]. This probability is given by

$$p(\mathbf{S}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\left(\sum_{n=1}^T \lambda^\top \Phi(s(n-1), s(n), \phi(\mathbf{X}, n))\right), \quad (1)$$

where $Z(\mathbf{X})$ is a normalization constant and λ is a weighting vector which is estimated in the training phase. In the training phase, we maximize the log-likelihood of the given sequence, given the features, using an L-BFGS algorithm [6]. For details and the training algorithm, see [1], [6], [8]. Φ is a vector-valued function which extracts information from the pure measurements with the use of the vector representations $\phi(\mathbf{X}, n)$ and combines this information with the state transition from state $s(n-1)$ to state $s(n)$.

CRFs are trained in an observed manner. That is, the state sequence \mathbf{S} is needed at least for the training data to apply the training [1], [3]. For the given setup, the data corresponds with time. Several activities are related to certain time frames, hence the features also occur at the specific time. To obtain a discrete-time model, we divide a day into K intervals of equal length and use this sequence for training. Each normal case state ζ_k with $k = 1, 2, \dots, K$ corresponds to a specific time. We have two classes of events: an event is either a strange measurement or a normal activity at the wrong time step.

A critical part of a CRF is the feature function Φ . This vector-valued function transforms the data into a feature space in which the CRF can separate the classes. Many applications use boolean feature transformations, for example [1]. We also use a feature extraction such that $\phi(\mathbf{X}, n)$ and $\Phi(s(n-1), s(n), \mathbf{X}, n)$ are sparse for any possible feature sequence \mathbf{X} . This simplifies the classification and event detection. We discuss this function in Section II-C.

C. Feature Function

It can be shown [1], [6], [8], [3] that we can apply CRFs using a feature function that processes the data such that each component of the vector is either 1 or 0. But CRFs are also capable of processing more general data [4]. In this paper, we propose a feature transformation that is feasible for CRFs and useful for the task of event detection.

Because the sensors do not measure their data at the same time, we do not have all data at every time step, hence we need a method to extract from this raw data a continuous sequence. For each minute, we integrate the data for intervals up to an hour. Hence, the time frame we integrate is a sliding window with a large overlap. This is a trade off between the treatment of rare and irregular sampling sensors and regular sampling sensors with high sampling frequency.

For an easier understanding of the feature transformation, assume that we have a one-dimensional signal $x(1), x(2), \dots, x(T)$ for which each sample $x(n)$ is an integrated time frame of the irregular sampled raw data. Because we use each dimension of the output signal individually, this is no loss of generality. In total, we compute L different features for each sensor. In the training phase, we observe a minimum value $x^{(min)}$ and a maximum value $x^{(max)}$. We add a margin δ_0 to those two values to define an upper and a lower bound, $x^{(max)} + \delta_0$ and $x^{(min)} - \delta_0$, respectively. In the experiments, δ_0 was chosen to be ten percent of the difference of $x^{(max)}$ and $x^{(min)}$. Moreover we divide the interval between those boundaries into $L - 2$ segments of equal length and test if an observation is in one of those segments. Further, we test if the observation exceeds one of the limits. This can be expressed as follows:

$$x^{(down)}(l) = x^{(min)} - \delta_0 + l \cdot \delta,$$

$$x^{(up)} = x^{(down)}(l) + \delta$$

with

$$\delta_0 = 0.1 \cdot (x^{(max)} - x^{(min)}),$$

$$\delta = \frac{x^{(max)} - x^{(min)} + 2\delta_0}{L - 2}.$$

Then we define $L - 2$ boolean features

$$\llbracket x^{(down)}(l) \leq x(n) \leq x^{(up)}(l) \rrbracket,$$

where the bracket $\llbracket P \rrbracket$ yields the value one if the predicate P is true and zero otherwise. Further, we define the features

$$\exp(x^{(min)} - \delta_0 - x(n)) \cdot \llbracket x(n) < x^{(min)} - \delta_0 \rrbracket - 0.1$$

and

$$\exp(x(n) - x^{(max)} + \delta_0) \cdot \llbracket x(n) > x^{(max)} + \delta_0 \rrbracket - 0.1.$$

All features from all sensors are stacked into a feature vector $\phi(\mathbf{X}, n)$.

Note that the last two features are negative if we observe the normal case. Hence, we gain negative weights λ_i for these features for all normal states. When a measurement exceeds the values observed in the training phase, these features become positive so the probability of the normal

states declines. In other words, the last two features allow us to measure an event in which the data exceeds its normal limits. However, this is not the only type of event we can detect. The probability for the normal case also becomes low when the measurements are within the given boundaries $x^{(min)} - \delta_0$ and $x^{(max)} + \delta_0$, but are strange for the actual state.

In the next step, the features are bound to states by a feature function Φ , and also the state transitions are included. The feature function Φ is defined as

$$\Phi(s(n-1), s(n), \mathbf{X}, n) = \begin{bmatrix} \llbracket s(n) = \zeta_k \rrbracket \cdot \phi(\mathbf{X}, n) \rrbracket_{k=1}^K \\ \left[\llbracket s(n-1) = \zeta_j \rrbracket \cdot \llbracket s(n) = \zeta_k \rrbracket \rrbracket_{j=1}^K \right]_{k=1}^K \\ \llbracket s(n) = \zeta_0 \rrbracket \cdot \phi(\mathbf{X}, n) \end{bmatrix}, \quad (2)$$

where K is the number of possible states $\zeta_1, \zeta_2, \dots, \zeta_K$ in the CRF. The state ζ_0 is a slack variable for observations which are not similar to the training features; it ensures the property $\sum_{i=0}^K p(s(n) = \zeta_i) = 1$, that is, the probability to either observe the normal case or an event is 1. For an example of a similar feature function, see [4].

D. Adding and Removing Sensors

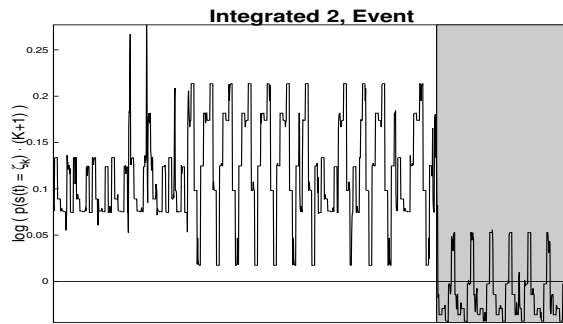
One of the benefits we gain by using CRFs instead of HMMs for the problem at hand is the possibility to simply and efficiently add and remove sensors, which can be challenging with other methods. As described in Section II-C, we first transform the observations and combine them into a feature vector. Because we do not alter the Markov model, the addition and removal of new features in the complete model is as follows.

If a sensor is removed from the sensor network temporarily (for example, if its battery status is too low), we set all the features that belong to this sensor to zero. By this and considering the definition of the CRF in (1) and the feature function in (2), we can see that in this case, the model still is appropriate. Of course, with less sensors we consider less information, which results in a lower significance of our method. If the sensor is removed permanently, we remove the corresponding values in the weighting vector λ and feature function.

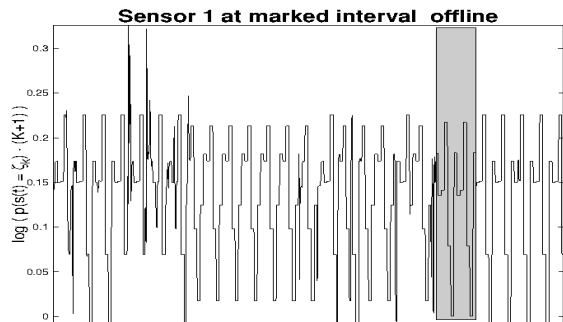
For the addition of sensors, we need to train the weights for new features only. The Markov model keeps constant and the other sensors are still online. Considering the CRF in (1) and the feature function in (2), the inclusion of the new sensors is to lengthen the feature function by the new features and the weights λ by the new weights only. A detailed description of this procedure for general CRFs can be found in [9].

III. EXPERIMENTS AND DISCUSSION

The algorithm is supposed to tackle two different targets. First, we demonstrate that the algorithm is suitable to detect events, second, we show that we can use the method for sensor networks which are not reliable, i.e. a sensor can be offline for periods of time.



(a) Event detected



(b) One sensor offline

Fig. 1. In (a) we can see that the probability of the actual state drops if we apply an event. In (b), we have removed a sensor in the marked time frame.

If we observe an event, this probability is supposed to decrease. If we deactivate a sensor, the probability is supposed to drop much less than if we observe an event.

For the events, however, we do not have representative data. Hence, we need to simulate the events. For this, we manipulate the data at given periods of time. As can be seen in Figure 1(b), the events can be detected.

For the offline sensors, we set the corresponding features to zero as described in Section II-D. As can be seen in Figure 1(a), the probability of the actual state decreases slightly as expected. The classification itself still is appropriate.

In total, we can detect the events at the correct time, and we can switch off some of the sensors without suffering a severe decrease of the power of the algorithm. Hence, we rate our experiments as a success.

IV. CONCLUSION AND OUTLOOK

We have presented a new algorithm for the detection of events, using a wireless sensor network in the context of AAL. The model has shown to be appropriate for this task, that is, we were able to determine the normal case and detect unknown events in the sequences. The method is very flexible and allows us to alter the number of devices in a simple manner. The experiments have shown that the CRF is capable to detect the true actual state from the given measurements. The integration feature extraction method has performed better than the interpolation in most cases.

The actual lack of events within the real test sequences

is a problem we cannot overcome by further processing. More importantly, we expect to detect real events in further experiments, because most of the patients hopefully stay healthy. So recorded events are very rare, and useful statistics can be applied on artificial events only. Since the study is ongoing and further data is being recorded in the households, real events may be included in future time.

In the future, we want to improve the Markov model we used in this paper. The first step is considering different time scales, such that a state does not represent measurements for minutes, but days and maybe weeks, and span the inference to analyze data of years. With this, we hope to detect slow, but pathologic changes.

Several extensions to the proposed algorithm are possible. For example, the CRF as we have introduced here can handle very different situations in a single state (at the same time slot, a patient can relax in his or her flat on one day and leave it for a walk another day), but we also wish to identify the subclasses. For this, we can use a similar model as in [7] to alter our model to include the subclasses.

For further interpretations, the method has to be applied in long-time studies. We can consider to implement high level interpretation of each state, sequences of states, and even include a certain event state as used in [4] to measure immediate events. We will also analyze additional sensors which can be included in the system. The analysis we have discussed in this paper can only be considered as a basis of the possibilities the system and especially the CRF based method for event detection may provide.

Acknowledgment This work has been supported by the Bundesministerium für Bildung und Forschung under Grant No. 16KT0942.

REFERENCES

- [1] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA, 2001, pp. 282–289. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645530.655813>
- [2] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, "Alarm-net: Wireless sensor networks for assisted-living and residential monitoring," Tech. Rep., 2006.
- [3] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00, San Francisco, CA, USA, 2000, pp. 591–598. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645529.658277>
- [4] D. Matern, A. P. Condurache, and A. Mertins, "Event detection using log-linear models for coronary contrast agent injections," in *Proceedings of the First International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, vol. 2, Vilamoura - Algarve, Portugal, 2012, pp. 172–179.
- [5] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Feb 1989, vol. 77, no. 2, pp. 257–286.
- [6] R. Gupta, "Conditional random fields," Technical report, IIT Bombay, 2006.
- [7] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *In NIPS*. MIT Press, 2004, pp. 1097–1104.
- [8] H. M. Wallach, "Conditional random fields: An introduction," University of Pennsylvania, Tech. Rep., 2004.
- [9] A. McCallum, "Efficiently inducing features of conditional random fields," in *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*, 2003. [Online]. Available: <http://citeseer.ist.psu.edu/mccallum03efficiently.html>