Semi-supervised Learning for Adaptation of Human Activity Recognition Classifier to the User

Božidara Cvetković, Mitja Luštrek, Boštjan Kaluža, Matjaž Gams
Jožef Stefan Institute, Department of Intelligent Systems
Jamova cesta 39, Ljubljana, Slovenia
{boza.cvetkovic, mitja.lustrek, bostjan.kaluza, matjaz.gams}@ijs.si

Abstract

The success of many ambient intelligence applications depends on accurate prediction of human activities. Since posture and movement characteristics are unique for each individual person, the adaptation of activity recognition is essential. This paper presents a method for on-line adaptation of activity recognition using semi-supervised learning. The method uses a generic classifier trained on five people to recognize general characteristics of all activities and a user-specific classifier briefly trained on the user using a reduced number of activities. The final decision on which classification to use for a given instance is done by a meta-classifier trained to decide which of the classifiers is more suitable for the classification. An instance classified with a sufficient confidence is added into the training set of the generic classifier. Experimental results show that the activity recognition accuracy increases by up to 11 percentage points with the proposed method. In comparison with Self-training proposed method performs better for up to five percentage points.

1 Introduction

Ambient intelligence (AmI) applications aim to provide relevant response to the human presence and have been widely researched and used in a variety of fields such as healthcare, eldercare, ambient assisted living, security, etc. Applications focused on user monitoring can benefit from efficient recognition of the activity in many ways. When the recognition is reliable the system can accurately detect deviations in the user's behavior, provide proper assistance and support in everyday life as well as adjust the environment and application to the user's habits, etc.

The most commonly used approach in activity recognition is supervised machine learning [Lester *et al.*, 2006]. Applications based on this approach are usually deployed with a generic classifier trained on the data collected in the laboratory environment and not on the behavior of the new end-user. In most cases once the system is trained and deployed it does not change anymore. The accuracy of activity recognition is thus affected by the difference in physical characteristics between the end-user and the people used in training. Consequently, the accuracy on real-life end-users with different characteristics may be substantially lower than in laboratory tests. Some approaches improve the activity recognition by using spatio-temporal information [Wu *et al.*, 2010].

The method we propose is trying to overcome the gap between end-users and the people used in training. This is achieved by employing two additional classifiers along with the generic classifier trained on general characteristics of the activities. The user-specific classifier is briefly trained during the initialization procedure on user specifics and the meta-classifier is trained to designate which of the activity recognition classifiers will label an instance. If the classification confidence value surpasses a specified threshold, the instance is added into the training set of the generic classifier. This method was deployed and validated in the project Confidence [2011], which uses a real-time localization system based on Ultra-wideband (UWB) technology with four wearable tags. The experimental results show that the activity recognition accuracy increases for up to 11 percentage points with the proposed method and in comparison with Self-training it performs better for up to 5 percentage points.

The paper is structured as follows. The related work on semi-supervised learning and adaptation of activity recognition is reviewed in Section 2. Section 3 introduces our experimental domain; Section 4 presents the proposed semi-supervised method and specifics of the learning procedures. In Section 5 we present the experimental results including method validation and comparison. Finally, Section 6 concludes the paper.

2 Related Work

Semi-supervised learning is a technique in machine learning that can use both labeled and unlabeled data. It is gaining popularity because the technology makes it increasingly easy to generate large datasets, whereas labeling still requires human effort, which is very expensive. The approach where the human annotator is required when the classifier is less confident in labeling is called Active learning [Settles, 2009]. Since in our case the human interaction is undesirable the Active learning approach is inappropriate, therefore we will focus on other semi-supervised learning techniques.

There are two categories of semi-supervised learning [Zhu, 2005]: single-classifier that use only one classifier and multiclassifier that use multiple classifiers, which can be split into multi-view and single-view approach. Key characteristic of a multi-view method is to utilize more feature independent classifiers on one classification problem. Single-view methods use classifiers with the same feature vector but differentiate considering the algorithm used for learning. We will review the techniques that relate to our proposed method.

The most common method that uses a single classifier is called Self-training. After an unlabeled instance is classified, the classifier returns a confidence in its own prediction, namely the class probability. If the class probability threshold is reached the instance is added to its training set and the classifier is retrained. The Self-training method has been successfully used on several domains such as handwriting word recognition [Frinken and Bunke, 2009], natural language processing [Guzmán-Cabrera *et al.*, 2008], protein-coding gene recognition [Guo and Zhang, 2006], etc.

Self-training was also applied to activity recognition by Bicocchi et al. [2008]. The initial activity recognition classifier was trained on the acceleration data and afterwards used to label the data from a video camera. The classified instances from the camera were added into the feature vector of the initial classifier and used for further activity recognition. This method can be used only if the initial classifier achieves high accuracy, since errors in confident predictions can decrease the classifier's accuracy.

Co-training [Blum and Mitchell, 1998] is a multi-view method with two independent classifiers. To achieve independence, the attributes are split into two feature subspaces, one for each classifier. The classifier that surpasses a confidence threshold for a given instance can classify the instance. The instance is afterwards added to the training set of the classifier that did not surpass the confidence threshold.

Democratic Co-learning [Zhou and Goldman, 2004] is a single-view technique with multiple classifiers. All the classifiers have the same set of attributes and are trained on the same labeled data with different algorithms. When an unlabeled instance enters the system, all the classifiers return their class prediction. The final prediction is based on the weighed majority vote among *n* learners. If the voting results returned 95% confidence or more the instance is added into the training set of all classifiers.

The modified multi-view Co-training algorithm called En-Co-training [Guan *et al.*, 2007] was used in the domain of activity recognition. The method uses information from 40 sensors, 20 sensors on each leg to identify the posture. The multi-view approach was changed into single-view by using all data for training three classifiers with the same feature vector and different learning algorithm which is similar to previously mentioned democratic Co-learning. The final decision on the classification is done by majority voting among three classifiers and the classified instance is added into the training set for all classifiers. This method improves the activity recognition; however the number of sensors is to high for unobtrusive system.

The method we propose is a single-view approach with two classifiers. Both are trained with the same algorithm but on different data. We use a third classifier to make the final prediction.

3 Confidence: A Brief Overview

The Confidence is an intelligent system for remote eldercare. The main objective is to detect deviations in short-term and long-term behavior of the end-user. There are currently three prototypes of the system in the verification phase in multiple European countries.

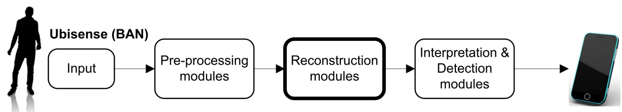

Figure 1: Simplified structure of the Confidence system. The method described in this paper is implemented as one of the reconstruction modules.

The simplified structure of the system is shown in Figure 1. The inputs to the system are the coordinates of four tags worn by the user. The coordinates are provided by the UWB real-time localization system Ubisense [Ubisense, 2010]. The user has a tag attached to the chest, waist and both ankles. The stated accuracy is approximately 15 cm but in practice larger deviations were observed.

The received data is sent to the pre-processing, where all four position coordinates are assembled into the current state in time denoted as snapshot. Each snapshot is processed by three filters. First, a median filter is applied, which eliminates large short-term changes in tag locations due to noise. Second, a filter that enforces anatomic constraints is used. This filter corrects errors such as an apparent lengthening of a limb. Third, the Kalman filter is applied, which smoothes sharp changes in both locations and speed.

The attributes for the recognition classifier are calculated from the filtered values. The attributes are the distances between the tags, velocity of the sensors and raw coordinates. For detailed explanation of the attributes the reader is referred to [Luštrek and Kaluža, 2009] where the authors used up to twelve tags to find appropriate attributes. The majority of the attributes computed by this module are for activity recognition by machine learning. The goal of activity recognition is to accurately identify the following eight human postures: lying, standing, sitting, falling, sitting on the ground, on all fours, going down and standing up.

The recognized activities serve as one of the inputs for the interpretation and detection modules focused on determining possible short-term or long-term behavior deviations [Luštrek *et al.*, 2009], that may indicate a health problem. Additional inputs are the characteristics of the user's movement, such as the speed of movement and various gait properties, and the user's location in the room with respect to the furniture (bed, chair). When a short-term deviation is detected, an alarm is raised and the output module issues a call for immediate help. When a long-term deviation is detected, a warning is sent describing the deviation, which may help a medical professional to determine whether it is a sign of an emerging disease.

Misclassification of the activity can result in a false positive alarm and in the worst case even in a false negative alarm, which can directly jeopardize the end-user's wellbeing. This shows that it is essential to accurately classify the activities in order to avoid such hazardous situations.

The main reason for misclassification of the activity, if we discard the noise, is the difference in the physical characteristics among users. The generic classifier employed in the system is trained on data of isolated set of people and does not contain the specific characteristics of the end-user. To overcome this problem we apply the method presented in the next section, which enables the system to learn the specifics of the user with semi-supervised learning.

4 The Adaptation Method

The Confidence system as well as other AmI systems that continuously monitor a user produce a large amount of unlabeled data for a particular end-user. These data are usually discarded, but they can be used to adapt the activity recognition classifier to the particular user.

We propose a method that adapts a system equipped with a generic classifier for activity recognition to a particular end-user.

The method consists of two steps:

- Initialization step
- On-line learning step

The initialization step is executed only once when the system is introduced to the end-user for the first time. During this process short labeled recordings of a subset of activities are made and used for training the user-specific classifier. The on-line learning operates in a non-supervised fashion, where both the user-specific and the generic classifiers are utilized for activity recognition. Activities classified with a sufficient confidence are used as additional training data for the generic classifier, which over time becomes adapted to the end-user. User-specific classifier is never retrained.

4.1 Initialization Step

The initialization step is performed only once at the beginning to introduce a new user to the system.

During this step the user is briefly recorded while performing basic activities that are defined in the recognition repertoire, namely standing, lying and sitting, since they are easy to perform. The transition activities such as falling, going down, standing up, sitting on the ground and on all fours are non-basic activities, since they are either uncomfortable to perform or very hard to label. The user is asked to perform each basic activity for a certain amount of time, in our case 60 seconds. During the recording procedure the captured data is labeled and used for the initial training of the new user-specific classifier.

The initialization step also involves modification of the generic classifier. The attributes related to the user's height are scaled by multiplying the value with the quotient of the user's height and the average height of the people used for generic classifier training. After the normalization the generic classifier is retrained.

The initialization step results in a new user-specific classifier and a modified generic classifier, both involved in the next step.

4.2 On-line Learning

The on-line learning step starts after the initialization and is performed until the stopping criterion is met, that is when the generic classifier is chosen to label most of the new istances.

The flow chart of the algorithm is presented in Figure 2. An unclassified instance is separately classified by two classifiers, the generic classifier and the user-specific classifier. Each of them returns the class distribution for the current instance. The meta-classifier decides which of the activity recognition classifiers is more likely to predict the class correctly. If the probability for the class returned by the chosen classifier surpasses a threshold, the instance is added to the training set of the generic classifier. In our case the threshold is 100%. After a period of time the generic classifier containing additional instances is retrained and thus adapted to the characteristics of the user. In our case we retrained the classifier every five minutes.

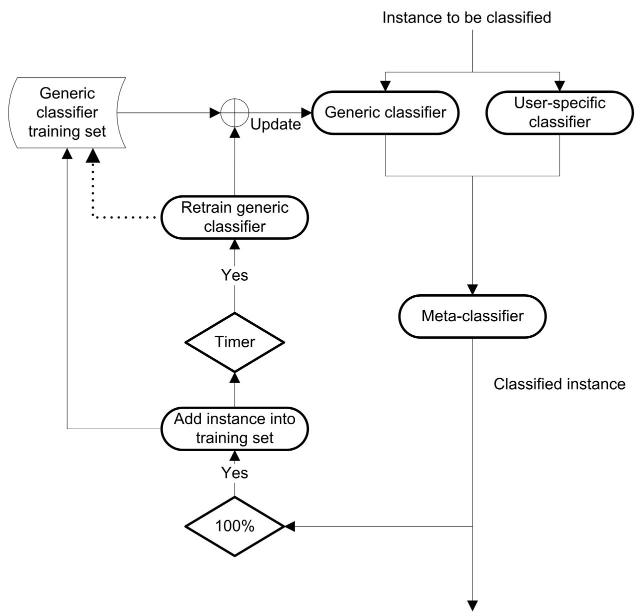

Figure 2: A work-flow of the on-line adaptation method.

To achieve a degree of balance between the classes and to add weight to the non-basic class instances added to the training set of the adapted generic classifier, the basic class instances are added only once, whereas the non-basic instances are added in triplicate. Adding only one or two instances resulted in slower learning. The reason for adding instances into the generic classifier and not the user-specific classifier is that the latter one is not equipped to handle all known activities, only those on which it was trained during the initialization step.

4.3 The Classifiers

The **Generic classifier** was build from the data we contributed to UCI Machine Learning Repository, under the title Localization Data for Person Activity, which was also used by Kaluža et al. [2010]. This dataset contains recordings of five

ML Algorithm	Attribute combination and accuracy %				
	Snapshot + Set 1	Set 1	Set 3	Set 1 + Set 3	**Set 1 + Set 2**
SVM	86.6	92.9	88.9	87.8	88.3
C4.5	96.8	95.4	96.1	96.6	95.9
Random Forest	90.9	95.9	96.6	96.9	**97.4**
Naive Bayes	61.0	75.7	70.1	68.8	82.3
AdaBoost	88.6	84.8	84.6	84.6	79.0
Bagging	96.9	94.7	95.8	96.2	95.8

Table 1: Attribute and machine learning algorithm combinations tested with the Meta-classifier.

people performing a scenario composed of eight activities: lying, standing, sitting, going down, standing up, sitting on the ground, on all fours and falling. The output of the generic classifier is the probability distribution over the classes corresponding to the eight activities given by Equation 1.

$$Pr_G = [Pr_G(C_1), \ldots, Pr_G(C_8)] \tag{1}$$

For validation of this classifier we used leave-one-person-out approach, where a classifier is built using the data of four persons and tested on the data of the fifth person. The classifier was trained using the Random Forest algorithm [Breiman, 2001] with attributes as described in Section 3. For the improvement of this classifier, we used the height of the end-user to scale the values of the height-related attributes. The scaled attributes are only the distances between the tags regarding the z-coordinate, since other attributes do not reflect the height. The measured accuracy was 86%.

The **User-specific classifier** is trained on the data recorded during the initialization procedure. Each posture is recorded for 60 seconds and given the sampling rate of 10 Hz we get approximately 1200 instances for the classifier training. This classifier was trained with the Random Forest algorithm. The feature vector is the same as in the generic classifier. The user-specific classifier is not able to recognize all activities. In our case it is trained to recognize basic activities: lying, standing and sitting; it has no knowledge about other activities. The output is the probability distribution over the eight classes given by Equation 2, where the unknown classes have zero probability, i.e. sitting on the ground, falling, on all fours, going down and standing up.

$$Pr_U = [Pr_U(C_1), \ldots, Pr_U(C_8)] \tag{2}$$

The **Meta-classifier** is used to determine the final activity of the current instance. It is trained before the system is deployed and is not adapted to the end-user. We compared the accuracy using several possible attribute sets for the meta-classifier. The results of the sets with best results are shown in the Table 1, where snapshot presents a current state of four tags.

The attributes in set 1 are represented by Equations from 3 to 8.

$$C_G = argmax_{i=1\ldots8}(Pr_G(C_i)) \tag{3}$$

$$C_U = argmax_{i=1\ldots8}(Pr_U(C_i)) \tag{4}$$

$$P_{GC_G} = Pr_G(C_G) \tag{5}$$

$$P_{UC_U} = Pr_U(C_U) \tag{6}$$

$$B_{CLASS} = \begin{cases} 1, & \text{if } C_i \in \{\text{standing, sitting, lying}\} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$EqualC = \begin{cases} 1, & \text{if } C_G = C_U \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The C_G and C_U represent the classification of the Generic and User-specific classifier, which are the classes with the highest probabilities in the class distribution. These probabilities are represented by P_{GC_G} and P_{UC_U}. The binary attribute B_{CLASS} tells whether the classification returned by the classifier selected by meta-classifier is a basic activity. The attribute represented by $EqualC$ tells whether the generic and user-specific classifier returned the same class.

Set 2 contains only the two attributes represented by Equations 9 and 10: the probability for the class selected by the user-specific classifier as computed by the generic classifier P_{GC_U} and the probability for the class selected by the generic classifier as computed by the user-specific classifier P_{UC_G}.

$$P_{GC_U} = Pr_G(C_U) \tag{9}$$

$$P_{UC_G} = Pr_U(C_G) \tag{10}$$

The attributes in set 3 are the z-coordinates of all tags, the distance between the chest and ankles and the distance between the chest and waist. Experiments showed that the distances in set 3 are not person-independent. Since the meta-classifier is not adapted to the end-user, these attributes had to be omitted.

The training of the meta-classifier was done on 60 minutes of labeled data of a person not used for further experiments. The data was collected from the recordings of a person performing a sequence of activities defined by the scenario. Each instance from the recording was passed over to the generic and user-specific classifier for classification. The class of the meta-classifier was defined according to the true class of the input instance and the relation to the prediction of the activity classifiers. We have tested all sensible combinations of the sets and the results five with the best results are shown in Table 1. The results show that the highest accuracy was achieved using attributes from the sets 1 and 2 and the Random Forest algorithm.

Activity Class	Person 1		Person 2		Person 3		Person 4	
	Start	End	Start	End	Start	End	Start	End
Lying	81.6	87.8	96.8	98.4	75.2	75.7	94.3	98.0
Standing	95.5	98.5	92.8	98.6	96.2	98.8	89.3	99.4
Sitting	35.9	80.1	88.7	99.1	52.2	76.5	75.0	97.7
Going down	52.0	52.9	42.7	54.6	51.8	55.4	16.7	12.8
Standing up	56.7	57.5	57.8	58.4	42.6	43.0	44.3	50.5
Sitting on the ground	28.8	63.4	22.0	40.2	83.3	86.6	46.5	36.1
On all fours	100	77.8	20.0	24.0	82.6	84.8	38.5	42.3
Falling	3.6	18.7	42.0	46.0	14.3	24.3	1.0	2.1
Overall	73.0	84.1	76.8	83.4	76.4	82.0	77.1	83.1

Table 2: The results of the on-line semi-supervised learning on four people. The results show the accuracies for each class and the overall accuracy (%) before the normalization and after the adaptation.

	Person			
	1	2	3	4
Difference in height (cm)	-18	-16	-5	+12
Starting accuracy (%)	73.0	76.8	76.4	77.1
Accuracy after normalization (%)	79.9	77.1	79.0	77.2
Accuracy after on-line adaptation (%)	84.1	83.4	82.0	83.1

Table 3: The difference in height per person according to the average height of the generic classifier, accuracy of the generic classifier before the adaptation process, increase in accuracy after normalization and the accuracy of the generic classifier after on-line adaptaion.

5 Experimental results

The method was integrated as one of the reconstruction modules in the Confidence system and was run on four different people with different physical characteristics. For the test set, every person performed the same sequence of activities defined in a scenario. The scenario captured typical daily activities during entire day, as well as some falls. A part of the scenario that represents the morning is for example lying in the bed, waking up, walking to the bathroom, sitting in the bathroom and falling in the bathroom. Each continuous sequence of the scenario lasted approximately 20 minutes and was repeated by the same person five times. Four of the recordings of each person were used for on-line learning and the final one to test the accuracy of the adapted classifier.

The experimental procedure was as follows: the system was initialized for the specific user (1 minute each basic activity), the user-specific classifier was trained and the generic classifier was normalized to the user's height. We learned in preliminary experiments that the scaling of all attributes for all instances can lead to higher noise for the activities taking place close to the ground. The misclassification happens because the lying activity is often classified as other activities where the z-coordinates of the chest and the waist are relatively close, for example on all fours. To avoid these types of misclassification we omitted the normalization of the lying instances. Attributes that are representing the distances between tags were selected for the normalization.

After the initialization process the on-line learning was started. The algorithm was run on four 20-minute recordings for each tested person and the accuracy of the adapted generic classifier was calculated every five minutes. The accuracy evaluation was done on the fifth recording of the person that was not used in the on-line learning procedure. The analysis of the progress of the adaptation process has shown that in the beginning all the instances added to the training set belonged to a basic class. During the fourth recording the majority of instances belonged to a non-basic class. In the beginning of the on-line learning the superior knowledge of the user-specific classifier was exploited to teach the generic classifier about the basic classes' specifics for the current user. As a consequence, later in the process generic classifier was more confident in the classification of the non-basic activities.

The results of the adapted generic classifier after the last processed recording are shown in Table 2. The table presents the accuracies of each class and the overall accuracy of the generic classifier before normalization and after the stopping criteria of the on-line learning was reached. The stopping criterion was reached in case the generic classifier classified all instances in the last 10 minutes.

The improvement of the generic classifier accuracy after normalization can be seen in Table 3. The table presents the difference in height regarding the average height of the people used in generic classifier, accuracy of the generic classifier before the process of adaptation started, accuracy of the generic classifier after normalization and accuracy of the adapted generic classifier. In the case of Persons 2 and 4 we see that normalization does not improve the generic classifier much and with the proposed method we can gain more than 5 percentage points of accuracy as seen in Table 2.

The proposed method was compared with the well known

Method	Gain in accuracy per person (pp)			
	1	2	3	4
Self-training	+8.63	+1.14	+2.08	+3.29
Proposed method	+11.10	+6.60	+5.60	+6.00
Difference	+2.47	+5.46	+3.52	+2.70

Table 4: The comparison between Self-training and our proposed method.

method for semi-supervised learning called Self-training. The results are presented in Table 4. We can observer that Self-training did increase the accuracy of the generic classifier, however our proposed method outperformed the Self-training by at least 2.47 percentage points and in best case by up to 5.46 percentage points.

6 Conclusion

This paper describes a method for on-line semi-supervised learning. The method uses generic, specific and meta-classifier. It was validated on the adaptation of the activity recognition. We showed that because of the difference in physical characteristics among the people, this method can be used to select informative instances in real-time and re-train the generic classifier to adapt it to a specific user. If we omit the gain in accuracy by simple height normalization, we can still show an increase in accuracy of 5 percentage points. The method was compared with Self-training method and the results showed that our proposed method outperformed it by 3.5% on average.

In thefuture the method should be compared with other known methods for semi-supervised learning and additionally verified on more people. To improve the method we will introduce a measure to balance the classes, since some of them have considerably more instances than others. For long-term use of the method it would be necessary to introduce aging of data. Finally, since this method has proven successful on our activity recognition domain, it should be tested on other domains as well.

Acknowledgments

The research leading to these results has received funding from the European Community's Framework Programme FP7/2007-2013 under grant agreement no 214986.

References

[Bicocchi *et al.*, 2008] Nicola Bicocchi, Marco Mamei, Andrea Prati, Rita Cucchiara, and Franco Zambonelli. Pervasive self-learning with multi-modal distributed sensors. In *Proceedings of the 2nd IEEE IWSOS*, pages 61–66, Washington, DC, USA, 2008. IEEE Computer Society.

[Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th COLT*, pages 92–100, New York, NY, USA, 1998. ACM.

[Breiman, 2001] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, October 2001.

[Confidence, 2011] Project Confidence. http://www.confidence-eu.org/, 2011.

[Frinken and Bunke, 2009] Volkmar Frinken and Horst Bunke. Self-training strategies for handwriting word recognition. In *Proceedings of the 9th ICDM*, pages 291–300, Berlin, Heidelberg, 2009. Springer-Verlag.

[Guan *et al.*, 2007] Donghai Guan, Weiwei Yuan, Young-Koo Lee, Andrey Gavrilov, and Sungyoung Lee. Activity recognition based on semi-supervised learning. In *Proceedings of the 13th IEEE RTCSA*, pages 469–475, Washington, DC, USA, 2007. IEEE Computer Society.

[Guo and Zhang, 2006] Feng-Biao Guo and Chun-Ting Zhang. Zcurvev: a new self-training system for recognizing protein-coding genes in viral and phage genomes. *BMC Bioinformatics*, 7(1):9, 2006.

[Guzmán-Cabrera *et al.*, 2008] Rafael Guzmán-Cabrera, Manuel Montes-Y-Gómez, Paolo Rosso, and Luis Villaseñor Pineda. A web-based self-training approach for authorship attribution. In *Proceedings of the 6th GoTAL*, pages 160–168, Berlin, Heidelberg, 2008. Springer-Verlag.

[Kaluža *et al.*, 2010] Boštjan Kaluža, Violeta Mirchevska, Erik Dovgan, Mitja Luštrek, and Matjaž Gams. An agent-based approach to care in independent living. In *AmI*, volume 6439 of *LNCS*, pages 177–186, Berlin, Heidelberg, 2010. Springer-Verlag.

[Lester *et al.*, 2006] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Pervasive Computing*, volume 3968, pages 1–16, Berlin, Heidelberg, 2006. Springer-Verlag.

[Luštrek and Kaluža, 2009] Mitja Luštrek and Boštjan Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):197–204, 2009.

[Luštrek *et al.*, 2009] Mitja Luštrek, Boštjan Kaluža, Erik Dovgan, Bogdan Pogorelc, and Matjaž Gams. Behavior analysis based on coordinates of body tags. In *AmI '09 Proceedings of the ECAmI*, pages 14–23, Berlin, Heidelberg, 2009. Springer-Verlag.

[Settles, 2009] Burr Settles. Active learning literature survey. CS Technical Report 1648, University of Wisconsin–Madison, 2009.

[Ubisense, 2010] Ubisense. http://www.ubisense.net, 2010.

[Wu *et al.*, 2010] Chen Wu, Amir Hossein Khalili, and Hamid Aghajan. Multiview activity recognition in smart homes with spatio-temporal features. In *Proceedings of the 4th ACM/IEEE ICDSC*, pages 142–149, New York, NY, USA, 2010. ACM.

[Zhou and Goldman, 2004] Yan Zhou and Sally Goldman. Democratic co-learning. In *Proceedings of the 16th IEEE ICTAI*, pages 594–202, Washington, DC, USA, 2004. IEEE Computer Society.

[Zhu, 2005] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, CS, University of Wisconsin-Madison, 2005.