Multi-Classifier Adaptive Training: Specialising an Activity Recognition Classifier Using Semi-Supervised Learning

Božidara Cvetković, Boštjan Kaluža, Mitja Luštrek, and Matjaž Gams

Jožef Stefan Institute,
Department of Intelligent Systems,
Jamova ceta 39, 1000 Ljubljana, Slovenia
{boza.cvetkovic,bostjan.kaluza,mitja.lustrek,matjaz.gams}@ijs.si
http://ijs.dis.si

Abstract. When an activity recognition classifier is deployed to be used with a particular user, its performance can often be improved by adapting it to that user. To improve the classifier, we propose a novel semi-supervised Multi-Classifier Adaptive Training algorithm (MCAT) that uses four classifiers. First, the General classifier is trained on the labelled data available before deployment. Second, the Specific classifier is trained on a limited amount of labelled data specific to the new user in the current environment. Third, a domain-independent meta-classifier decides whether to classify a new instance with the General or Specific classifier. Fourth, another meta-classifier decides whether to include the new instance in the training set for the General classifier. The General classifier is periodically retrained, gradually adapting to the new user in the new environment where it is deployed. The results show that our new algorithm outperforms competing approaches and increases the accuracy of the initial activity recognition classifier by 12.66 percentage points on average.

Keywords: semi-supervised learning, adaptation to the user, MCAT, activity recognition

1 Introduction

Activity recognition applications are often faced with the problem that a classifier trained in a controlled environment can demonstrate a high accuracy when tested in such environment, but a drastically lower one when deployed in a real-life situation. This issue could be resolved if one were to label enough data specific for the user in the real-life situation where the classifier is deployed. However, since this is often unpractical, semi-supervised learning can be employed to label the data of real users automatically. In semi-supervised learning, one or multiple classifiers usually classify each instance, some mechanism selects the final class based on their outputs, and if the confidence in this class is sufficient, the class is used to label the instance and add it to the training set of the classifiers. This approach raises three challenges that need to be addressed: (i) how to design the multiple classifiers and what data to use for training each of them; (ii) how to choose the final class; and (iii) how to decide whether an instance will be added to the training set of the classifiers.

This paper presents a novel algorithm for adaptation of an activity recognition classifier to a user called Multi-Classifier Adaptive Training (MCAT). It addresses all the above-mentioned challenges. The algorithm is based on the following key contributions: (i) it introduces two classifiers for labelling, a General one, which is trained on the general activity recognition domain knowledge, and a Specific one, which is trained on a limited number of labelled instances specific for the current end-user; (ii) the selection of the final class is handled by a meta-classifier, which uses the specific knowledge to improve the general knowledge of the activity recognition domain; and (iii) the decision about which instance to include in the training set is tackled with another meta-classifier, which weighs the decision of the first meta-classifier. The two meta-classifiers allow us to combine the general knowledge of the activity recognition domain with the specific knowledge of the end-user in the new environment in which the classifiers are deployed. The final contribution of the paper is the training procedure, which uses all the available data to properly train each of the four classifiers.

The algorithm was implemented and evaluated on an activity recognition domain based on Ultra-Wideband (UWB) localisation technology. The people had four wearable sensors attached to their clothes (chest, waist, left ankle, right ankle). The general activity recognition domain knowledge was induced from the data of people performing activities wearing the sensors in the laboratory. The specific data was obtained from an individual person to whom the system is adapted. The proposed approach is compared to two non-adaptive approaches and to three adaptive approaches: the initial version of the proposed approach [1], the self-learning algorithm [2] and majority vote algorithm [3, 4].

The results show that the MCAT method successfully increases the accuracy of the initial activity recognition classifier and significantly outperforms all three compared methods. The highest absolute increase in accuracy, when using the MCAT method, is by 20.58 percentage points.

The rest of the paper is structured as follows. Section 2 contains the related work on the semi-supervised learning approaches and their use in adaptation of the activity recognition. The motivating domain is introduced in Section 3. The MCAT method is explained in the Section 4 and Section 5 contains the experiment and the results. Section 6 concludes the paper.

2 Related Work

Semi-supervised learning is a technique in machine learning that uses both labelled and unlabelled data. It is gaining popularity because the technology makes it increasingly easy to generate large datasets, whereas labelling still requires human effort, which is very expensive. The main idea of the semi-supervised approach is to use either i) supervised learning to label unlabelled data or ii) to utilise additional labelled data with unsupervised learning.

A similar approach to semi-supervised learning is active learning. This approach uses supervised learning for the initial classification. However, when the classifier is less confident in labelling the human annotator is required [5, 6]. Since human interaction is undesirable the Active learning is inappropriate.

There are four ways of categorising semi-supervised learning approaches [2]: (i) single-classifier and multiple-classifier; (ii) multi-view and single-view; (iii) inductive and transductive; and (iv) classifier- and database-based approaches. The single classifier methods use only one classifier for classification task, where multiple classifiers use two or more classifiers. The key characteristic of a multi-view method is to utilise multiple feature-independent classifiers in one classification problem. Single-view methods use classifiers with the same feature vector but differentiate in the algorithm used for learning. Inductive methods are those that first produce labels for unlabelled data and secondly a classifier. Transductive methods only produce labels and don't generate a new classifier. Classifier-based approaches start from one or more initial classifiers and enhances them iteratively. The database-based approaches discover an inherited geometry in the data, and exploits it to find a good classifier. In this paper we will focus on a single-view, inductive and classifier-based semi-supervised learning.

The most common method that uses a single classifier is called self-training [2]. After an unlabelled 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 applied to activity recognition by Bicocchi et al. [7]. The initial activity recognition classifier was trained on the acceleration data and afterwards used to label the data from a video camera. This self-training method can be used only if the initial classifier achieves high accuracy, since the errors in confident predictions can decrease the classifier's accuracy. The self-training has also been successfully applied on several other domains such as handwriting word recognition [8], natural language processing [9], protein-coding gene recognition [10].

Co-training [11] 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. A major problem of this algorithm is that the feature space of the data cannot always be divided orthogonally. If the attributes are split at random it is possible that classifiers do not satisfy the requirement of self-sufficiency.

The modified multi-view Co-training algorithm called En-Co-training [12] 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 the data for training three classifiers with the same feature vector and a 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.

Democratic co-learning [13] is a single-view technique with multiple classifiers. All the classifiers have the same set of attributes and are trained on the same labelled data but with different algorithms. When an unlabelled instance enters the system, all the classifiers return their class prediction. The final prediction is based on the weighed majority vote among the classifiers. If the voting results return sufficient confidence, the instance is added into the training set of all the classifiers.

The MCAT method uses two classifiers; both are trained with the same algorithm but on different data. We use a third, meta classifier, to make the final prediction on the class. The decision whether to put an instance into the training set or not is solved by employing another meta classifier and not slightly arbitrarily like in the case of the Democratic co-learning. In contrast to Co-training, our two domain classifiers have the same attribute set, thus the problem of dividing the sets is not present.

3 Motivating Domain

In this paper the MCAT method is applied to activity recognition, a very common domain in ambient intelligence. Activity recognition classifier is usually trained on the data retrieved from the people performing activities in a controlled environment such as a research laboratory. The classifier trained in this fashion can perform with high accuracy when tested in the same environment, but it is likely that it will perform poorly when deployed in a new environment with a new person, since each person tends to have specific manner of performing the activities. We have faced this problem during the development of the Confidence system [14][15].

The Confidence system is a real-time system developed for constant monitoring of human activity and detection of abnormal events such as falls, or unusual behaviour that can be a result of a developing disease. It is based on a multi-agent architecture where each module, task or activity is designed as an agent providing a service. The multi-agent architecture is shown in Figure 1. It consists of seven agent groups: (i) sensor agents that serve raw localisation data to the next group of agents, (ii) refining agents that filter out the noise, (iii) reconstruction agents that determine the location and activity of the user and are the main trigger for other agents, (iv) interpretation agents that try to interpret the state of the user and provide the emergency state information, (v) prevention agents that observe the user and detect possible deviations in the behaviour, (vi) cognitive agents that monitor the cognitive state of the user, and (vii) communication agents dedicated to user interaction in case of emergency. For detailed description of the multi-agent architecture of the Confidence system the reader is referred to [16].

The input to the system or the sensor agents are the coordinates of the Ubisense location tags [17] attached to the user's waist, chest and both ankles. Since the Ubisense system is noisy, we use three filters, implemented in the refinement agents group, to reduce it. First the data is processed by Median filter, secondly the data is processed with Anatomic filter, which applies anatomic constraints imposed by the human body and the last filter is the Kalman filter. For more details on the filters the reader is refereed to [18]. To get a representation for all the tag positions in time, the snapshots are created with frequency of 4 Hz. Each snapshot is augmented with positions and number of attributes, which are used for activity recognition and other purposes. Primarily the activity recognition was developed to enhance the fall detection, therefore the accuracy of the activity recognition is crucial. The reader is referred to [19] for details about the used attributes.

The activity recognition agent is included in the reconstruction agents group and is able to recognise nine atomic activities: standing, sitting, lying, falling, on all fours, sitting on the ground, standing up and sitting down. The high accuracy of the activity recognition is an essential information to be passed to the other agents for further effective reasoning about the abnormal events. Adapting the activity recognition agent to each user helps achieve that. Our method that does exactly that is described in the next section.

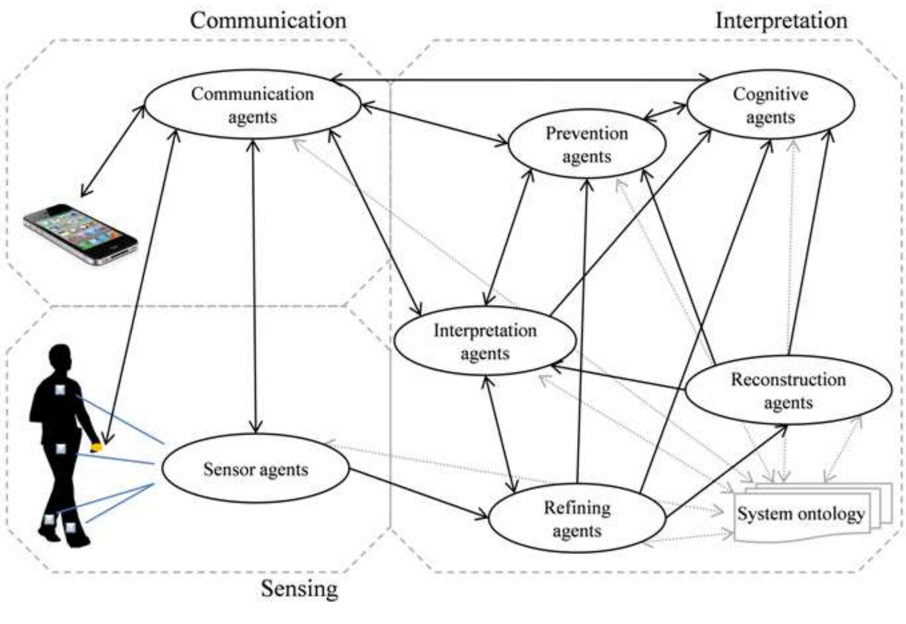

Fig. 1. The multi-agent system architecture of the Confidence system.

4 The Multi-Classifier Adaptive Training Method (MCAT)

In this section we propose the MCAT method that improves the classification accuracy of an initial activity recognition classifier utilising unlabelled data and auxiliary classifiers. Before deploying the classifier in a real-world environment with the new user, it is usually trained in a controlled environment on a limited amount of data. In addition to this general approach, a small amount of labelled data from the new real-world environment and a new user is obtained. The MCAT method uses the data from the new environment to adapt the initial classifier to the specifics of the environment while using it.

Consider the following example: The initial classifier is trained on activities performed by several people. When using this classifier on a new person whose physical characteristics are different, and who was not used for training, the recognition accuracy can be low, since each person has specific movement signature. The MCAT method utilises a few activities performed by the new person to learn his/her specifics and thus improve upon the classification of the initial classifier.

4.1 The Algorithm

The proposed MCAT algorithm is shown in Figure 2. The General classifier is the initial classifier trained on the general domain knowledge in the controlled environment. This would be the only classifier deployed in a typical application that does not use the proposed method. To improve the General classifier, we propose a set of three auxiliary classifiers: (i) the Specific classifier, which is trained on a small subset of manually labelled data; (ii) the Meta-A classifier, which decides which classifier (Specific or General) will classify an instance; and (iii) the Meta-B classifier, which decides whether the instance should be included in the training set of the General classifier.

The *General classifier* is trained on a general set of labelled data available for the activity recognition domain, i.e., a controlled environment. The attributes are domain-specific and the machine-learning algorithm is chosen based on its performance on the domain.

The *Specific classifier* is trained on a limited amount of labelled data specific for the new environment in which the classifiers are deployed (new person). Note that this limited dataset does not necessary contain all the classes that are present in the dataset of the General classifier. This may happen due to an unbalanced distribution of class labels, for example, in the activity-recognition domain quick and short movements such as falls are rare. The classes known to the Specific classifier are denoted as basic classes. The attributes and the machine-learning algorithm should preferably be the same as those used in the construction of the General classifier.

After both classifiers return their classes, the *Meta-A classifier* is activated. The decision problem of the Meta-A classifier is to select one of the classifiers to

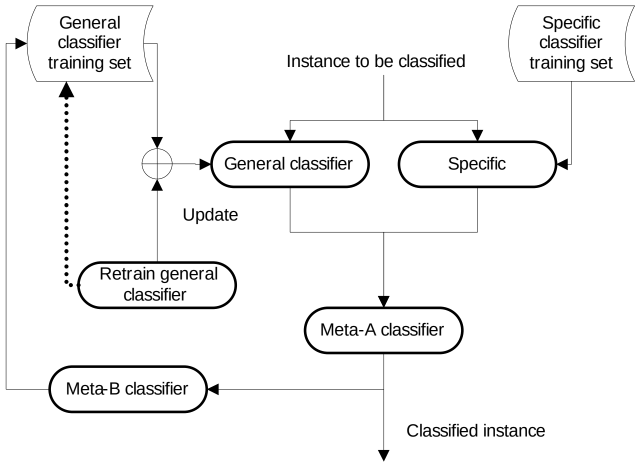

Fig. 2. The work flow of the algorithm proceeds as follows. The method contains two activity recognition classifiers, the General and Specific, and two additional meta-classifiers. The Meta-A classifier decides on the final class of the instance and the Meta-B classifier decides whether the instance is to be included in the training set of the General classifier.

classify a new instance. The Meta-A classifier can be trained with an arbitrary machine-learning algorithm. The attributes for the Meta-A classifier should describe the outputs of the General and Specific classifier as completely as possible, while remaining domain-independent. If we add domain attributes to the meta-attributes, the decision of the Meta-A classifier will also be based on the specifics of the training data available prior to the deployment of the classifiers, which may be different from the specifics of the situation in which the classifiers are deployed. It was experimentally shown in previous work [1] that domain attributes do not contribute to higher accuracy of the classifier, on the contrary, they can result in over-fitting to the specifics of the training data.

Although the attributes in two Meta-A classifiers, deployed in two different systems, need not be exactly the same, we can propose the following set of attributes considering the previous research: the class predicted by the generic classifier (C_G), the class predicted by the Specific classifier (C_S), the probability assigned by the generic classifier to C_G, the probability assigned by the Specific classifier to C_S, is the C_G one of the basic classes, are the classes C_G and C_S equal, the probability assigned by the generic classifier to C_S, the probability assigned by the Specific classifier to C_G. For more information on the tested sets of attributes and algorithms the reader is referred to [1].

The *Meta-B classifier* is used to solve the problem of whether an instance should be included in the General classifier's training set. The output of the Meta-B classifier should answer the question whether the current instance contributes to a higher accuracy of the General classifier. This question is not trivial and there are several approaches that specifically address it [20]. We use a heuristic instead, which answers the question: "Did the Meta-A classifier select the correct class for the current instance?". The heuristic performs well and is computationally inexpensive, so we left the investigation of more complex approaches for future work. The attributes used in the Meta-B classifier are the same as those used in the Meta-A classifier, with one addition: the confidence of Meta-A classifier in its prediction. The Meta-B classifier can be trained with an arbitrary machine-learning algorithm.

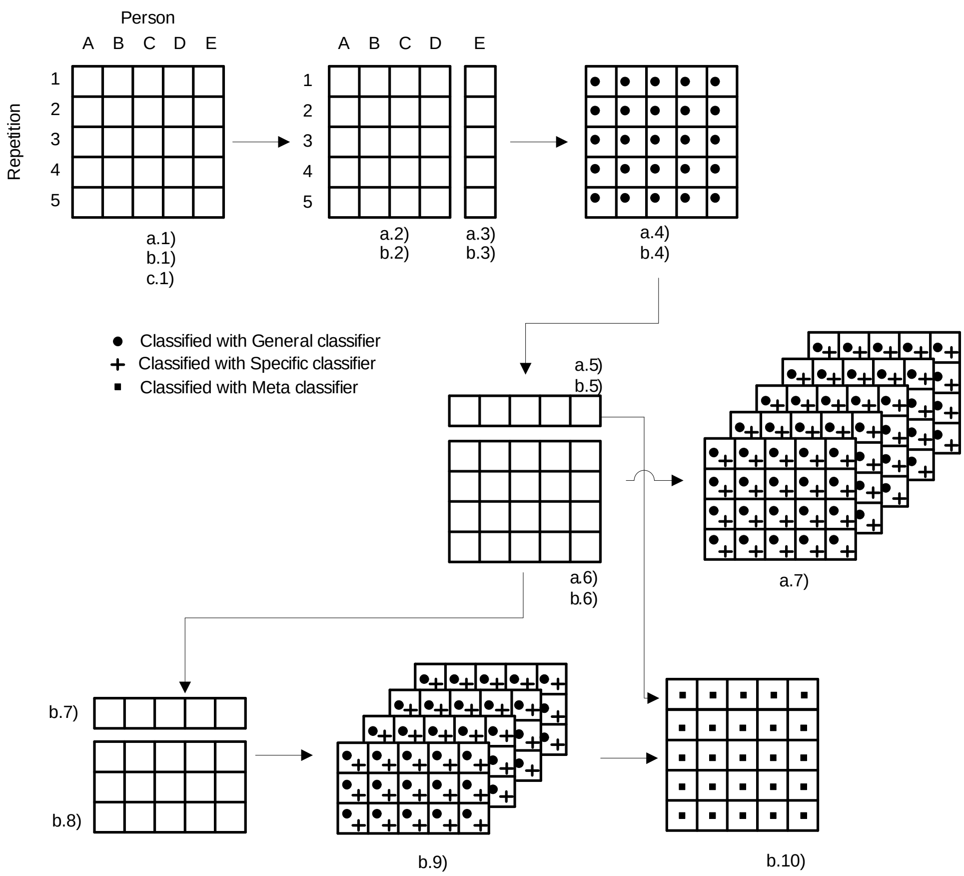

Fig. 3. Training of the classifiers. The figure shows three steps (a, b and c), each resulting in the training set for an individual classifier. *Step a* results in the training set for the Meta-A, *step b* results in the training set for Meta-B and *step c* in the training set for the General classifier. Each step is composed of sub- steps marked with numbers.

4.2 Training Procedure

Training of the four classifiers requires the data to be separated into non-overlapping datasets. This means that the data used to train the Meta-A classifier must not be used to train the General or Specific classifier, and the data used to train the Meta-B classifier must not be used to train these two or the Meta-A classifier. Since we are usually provided with a limited amount of data, our efforts must be focused on maximally utilising all the data.

We propose a training procedure that divides the data into several sets for training both meta-classifiers and the General classifier. The procedure consists of three steps as shown in Figure 3. The steps are marked with a, b and c, each consisting of several sub-steps. The first step trains the Meta-A classifier, marked with sub-steps a. It requires data classified by both domain classifiers (the General and Specific). In the second step the Meta-B classifier is trained, marked with sub-steps b. It requires data classified by the Meta-A classifier and data classified by both domain classifiers. The third step includes training of the Generic classifier, marked with step c. It requires only the originally labelled data.

Our procedure is general, but for the purpose of this paper we will make use of the activity-recognition domain. Specifically the publicly available dataset "Localization Data For Person Activity" from the UCI Machine Learning Repository [21]. The dataset consists of recordings of five people performing predefined scenarios five times. Each person had four Ubisense [17] tags attached to the body (neck, waist, left ankle and right ankle) each returning its current position. The goal is to assign one of eleven activities to each time frame. The dataset can be divided by the person five times and by the scenario repetition five times (sub-steps a.1,b.1,c.1).

The first step is building a training set for the Meta-A classifier (sub-steps a). Four persons are selected for training the General classifier (sub-step a.2), which is used to classify the fifth person (sub-step a.3). This is repeated five times for each person and the result is the complete dataset classified with the General classifier (sub-step a.4). The data classified with the General classifier is represented with dots in Figure 3. The data classified with the General classifier is split by repetition five times. Since the specific classifier should be trained on a small amount of data, one repetition for each person is used for building the Specific classifier (sub step a.5) and the remaining four repetitions are classified with the Specific classifier (sub step a.6). The data classified with the Specific classifier is represented with crosses in Figure 3. In total, we have five Specific classifiers (one per person), each of which was used to classify four repetitions. This gives us the data in the sub-step a.7, which represents the training set for the Meta-A classifier.

The second step is building the training set for the Meta-B classifier (sub-steps b). The sub-steps from b.1 to b.4 are executed identically as sub-steps from a.1 to a.4 as explained earlier. The data classified with the General classifier (dotted data in sub-step b.4) is divided by the repetition. One repetition for each person is kept aside for the Meta-B training (sub step b.5). The data that is left contains five people times four repetitions (sub-step b.6). It is used for the temporary Meta-A training. One repetition for each person is used to train the Specific classifier (sub-step b.7). The three repetitions that are left (sub-step b.8) are classified with the General and Specific classifier (sub-step b.9) represented with dots and crosses. The result is used to train the temporary Meta-A classifier. The data that was kept aside for the Meta-B classifier is now retrieved and classified with the temporary Meta-A classifier. The results are stored into the training set for the Meta-B classifier (sub step b.10). After all the repetitions are executed we get the training set (represented with squares) for training the final Meta-B classifier.

The last step is building the dataset for training the General classifier, which is trained on the complete dataset as shown in step c.1.

5 Experimental Evaluation

The experimental evaluation is focused on the activity recognition discussed in the previous sections. The main reason why the general classifier will likely not perform well in a real-life environment on a particular person is that each person has its own specifics such as the height and the characteristics of movement. The general classifier, which is trained on several people, can thus recognise the activities of a "general person". Obtaining enough training data for a particular end user is difficult, so our method is well suited for solving this problem.

5.1 Experimental Setup

The experimental data was collected using the real-time Confidence system described in Section 3. We have collected two different sets of data, one for the classifier training, namely *the training dataset* and one for testing the semi-supervised adaptation to the users, namely the *test dataset*.

The first dataset, *the training dataset* was contributed to the UCI Machine Learning Repository [21]. These data consists of the recordings of five people performing a scenario five times. The scenario is a recording of a person performing a continuous sequence of activities that represents typical daily activities. The data was divided into segments as discussed in Section 4.2 and used for training the classifiers.

The second, the *test dataset* consists of the recordings of ten people, again performing typical daily activities. All the people are different than in the *training dataset*. The scenario was repeated by each person five times, which gives us 2.7 hours of data per person on average, and 30.2 hours altogether. The scenario was designed to reflect the distribution of the activities during one day of an average person. The scenario contains eight activities: standing, lying, sitting, going down, standing up, falling, on all fours and sitting on the ground. The scenario performed by the people in the *test dataset* contains additional repetition of the sitting on the ground activity with all the respective transitions from sitting and afterwards to walking compared to the *training dataset*. Both datasets were primarily recorded for the purpose of fall detection and the difference between them is not significant for our problem.

The *training dataset* contains more labels that the test data, so we had to merge the transition activities. The activities "lying down" and "sitting down" were merged into the activity "going down". The activities "standing up from lying", "standing up from sitting" and "standing up from sitting on the ground" were merged into the activity "standing up". The activity "walking" was merged into "standing". The distributions of the activities per merged class for both datasets are shown in Table 1.

Table 1. Activities and their distributions per dataset.

	Distributions (%)	
Activities	Training dataset	Test dataset
Standing	19.56	29.55
Sitting	16.67	15.53
Lying	33.14	31.58
Sitting on the ground	7.20	16.09
On all fours	3.05	0.59
Falling	1.81	0.89
Going down	4.80	1.26
Standing up	13.75	4.50

To create datasets needed for the proposed method we divided the *test dataset* as follows: *Basic activity dataset* - ten people performing basic activities (exactly 30 seconds per activity), which are lying, standing and sitting and *Unlabelled test dataset* - ten people performing scenario five times (125 minutes per person on average). The *Basic activity dataset* is used to train the Specific classifier per person. The *Unlabelled test dataset* per person is the data used for the semi-supervised adaptation, the core of MCAT.

The experiment was done in two steps: first, the classifier training; and second, the use of MCAT as a semi-supervised adaptation of the initial General classifier.

The classifier training was executed as presented in Section 4.2. The Meta-A classifier was trained using the Support Vector Machines algorithm [22] and the Meta-B classifier using the C4.5 machine learning algorithm [23]. Note that the classifiers could be trained using other machine learning algorithms, but the chosen two algorithms proved satisfying in our experiments. Both were used with default parameters as implemented in the Weka suite [24]. The General classifier was trained using the Random Forest algorithm [25]. The accuracy of the classifier is 86% when evaluated using leave-one-person out approach, which implies that one can expect the same accuracy when the classifier is deployed in the new environment; however, this is not the case as we see in the Table

2 in the column labelled with G (initial general classifier). This algorithm was selected according to the results from previous research [26].

For each test person the Specific classifier was trained using the Random Forest algorithm on the *Basic activity dataset*. The *Unlabelled dataset* of that person was processed with the MCAT method. The General classifier was re-trained after each repetition of the test scenario (five repetitions per person), to take advantage of the instances from the unlabelled dataset that were included in its training dataset.

The data for each person was processed by our method two times. In the first run the instances selected for the inclusion in the training set of the General classifier were weighed with the weight 2. In the second run the weight was 1. The reason for increasing the weight in the first run is to accelerate the adaptation. The experiments showed that in case the weight in the first run is set to the default value 1, the number of non-basic class instances, which are sitting on the ground, falling, standing up, going down, on all fours, selected for the inclusion in the second run is lower. The weight value in the second run is decreased to avoid the elimination of the general knowledge from the General classifier. If an instance already existed in the training set and was selected for the inclusion again, it was discarded.

The MCAT method was compared to five competing methods: two transductive approaches that only label the instance and do not perform the adaptation; and three inductive semi-supervised learning approaches. The transductive approaches are: *the baseline approach*, which first merges the training data of the Generic and the Specific classifier into one training set and then trains a new General classifier (G'); and the *MCAT without Meta-B*(without semi-supervised learning) approach that builds both the Generic and Specific classifier and uses the Meta-A to decide which one to trust.

The inductive semi-supervised learning approaches used for comparison are *self-training*, *majority vote* and *threshold-based MCAT*. The self-training method uses the General classifier for classification. The instances in the *Unlabelled dataset* with the 100% classification confidence are added to its training set. The majority vote uses three classifiers trained on the training set with different machine learning algorithms. We used the algorithms that achieved the highest accuracy in the General classifier evaluation using the leave-one-person out approach. These were the Support Vector Machines, Random Forest and C4.5. The instances in the *Unlabelled dataset* with 100% classification confidence were included in the training set of all classifiers. The threshold-based MCAT uses threshold rule instead of the Meta-B classifier, which selects the instance to be included in the training set of the Generic classifier.

5.2 Results

The classification accuracies of the methods are shown in Table 2. The left side of the table shows the accuracy of the initial General classifier G and the Specific classifier S. The right side of the table shows the gain/loss in accuracy of the methods compared to the initial General classifier. The compared methods are:

(i) transductive approaches: baseline method (G&S merged); MCAT without Meta-B (Meta-A) and (ii) semi-supervised learning approaches: threshold-based MCAT (previous version PV); Self-training (ST) and Majority vote (MV) algorithm. The last column shows the results of the MCAT method.

The results for the General classifier show a decrease in accuracy when used in the different environment than the controlled where the accuracy was 86%. The average accuracy on ten people is 70.03% (Table 2, column *Initial G*). The results of the Specific classifier (Table 2, column *Specific S*) show that it achieves a higher accuracy than the initial General classifier overall and in several individual cases, even though it is able to predict only three basic classes (lying, standing, sitting).

The results on the baseline training set are the worst compared to the General classifier accuracy for four of the people. This is because some instances from the Specific and General classifier are similar but differ in the label. The results of using the Meta-A classifier to select the final class show that this method outperforms the General classifier by 6.14%. The higher accuracy is gained because of the knowledge of basic classes, which represent 76.67% of the dataset on average and by 16.81 percentage points in the best case (Person I). This increase in accuracy reveals that using a classifier for class selection in one of the reasons for the success of the proposed approach, while the other reason is the semi-supervised adaptation. The average accuracy of the General classifier after adaptation using the previous version is 78.59% and the average gain in accuracy is 8.56 percentage points. The results of the Self-training show that in a few cases, where the General classifier has low initial accuracy, the method performs poorly and further weakens the classifier. The average accuracy after adaptation is 71.91%. The results of the Majority vote method show that introducing extra classifiers contributes to gain in accuracy upon the initial classifier. The average accuracy after adaptation using the majority vote method is 74.29%.

The results for the MCAT method show the average gain in accuracy of 12.66 percentage points and the average accuracy of the classifier 82.70%. In the best case (Person C) it achieves an increase in accuracy of 20.58 percentage points and in the worst case (Person B) 7.38 percentage points. The highest absolute accuracy is achieved for Person J, 85.32%, and Person A, 84.46%, where both the General and the Specific classifier returned a similar initial accuracy. The MCAT method outperformed the self-training method by 10.79 percentage points, the majority vote by 8.41 percentage points and the previous version by 4.11 percentage points.

6 Conclusion

This paper is focused on the problem of enhancing an initial activity recognition classifier using unlabelled data. The main contributions of this paper are: (i) a novel method for specialising the activity recognition classifier by semi-supervised learning MCAT; and (ii) a procedure for training multiple classifiers from a limited amount of labelled data that fully utilises the data. The MCAT

Table 2. Classifier accuracies and comparison of the MCAT method to the (i) transductive approaches: baseline (G and S merged G&S), MCAT without adaptation (Meta-A) and (ii) semi-supervised approaches: threshold-based MCAT (previous version PV), Self-training (ST) and Majority vote (MV) method.

	Classifier Accuracy(%)		Method Comparison (gain/loss in pp against G)					
Person	Initial G	Specific S	G&S	Meta-A	PV	ST	MV	MCAT
A	75.28	76.82	-10.57	+3.62	+3.82	+2.36	+0.38	+9.18
B	76.28	60.06	+1.17	+0.58	+0.70	+0.28	+0.64	+7.38
C	62.87	70.85	+8.52	+12.82	+13.46	-1.18	+3.07	+20.58
D	69.55	76.17	-0.21	+8.99	+9.77	+2.23	+2.10	+12.57
E	68.13	74.23	+0.20	+5.78	+9.76	-1.81	+6.73	+16.22
F	73.57	68.18	-4.42	+2.62	+8.08	+6.07	+3.67	+8.86
G	65.42	67.72	5.97	+6.99	+9.76	-0.21	+8.57	+12.60
H	73.45	67.46	-8.02	+0.01	+4.51	+4.31	+0.41	+10.53
I	62.09	68.08	+7.75	+16.81	+16.98	-0.18	+11.03	+17.08
J	73.67	74.17	+1.18	+3.15	+8.73	+6.87	+5.98	+11.65
Average	70.03	70.37	+0.16	+6.14	+8.56	+1.87	+4.26	+12.66

method for the semi-supervised learning uses auxiliary classifiers to improve the accuracy of the initial General classifier. The method utilises a classifier trained on a small amount of labelled data specific for the current person, in addition to the general-knowledge classifier. The two additional meta-classifiers decide whether to trust the General or the Specific classifier on a given instance, and whether the instance should be added into the training set of the General classifier or not.

The MCAT method was compared to two transductive approaches and three inductive semi-supervised approaches. The MCAT method significantly outperforms both the baseline approach by 12.51 percentage points and the MCAT without Meta-B by 6.53 percentage points on average. The MCAT method also significantly outperforms inductive approaches: the self-training by 10.79 percentage, the majority vote by 8.41 percentage points, and the threshold-based MCAT by 4.11 percentage points on average. On average, the initial classifier is improved by 12.66 percentage points.

This method can significantly contribute to further development of the ambient intelligence applications, since many of them rely on recognition of human activity. Accurate recognition of atomic activities can also contribute to more reliable recognition of the complex activities.

Acknowledgments. The research leading to these results was partly supported by the ARRS under the Research Programme P2–0209, partly by the European Community's Framework Programme FP7/2007-2013 under grant agreement N 214986.

References

1. Cvetković, B., Kaluža, B., Luštrek, M., Gams, M: Semi-supervised Learning for Adaptation of Human Activity Recognition Classifier to the User. In: IJCAI 2011, Workshop on Space, Time and Ambient Intelligence, pp. 24–29. Barcelona (2011)
2. Zhu, X.:Semi-Supervised Learning Literature Survey. CS Technical Report, University of Wisconsin-Madison (2005)
3. Kuncheva, L. I., Whitaker, C. J., Duin R. P. W., Limits on the majority vote accuracy in classifier fusion. Pattern Analysis and Applications, 8, 22–31 (2003)
4. Longstaff, B., Reddy, S., Estrin, D.: Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In: 4th International Conference on Pervasive Computing Technologies for Healthcare, pp. 1–7, IEEE press (2010)
5. Dasgupta, S.: Two faces of active learning. Theoretical Computer Science, 412, 1767–1781 (2011)
6. Settles, B.: Active Learning Literature Survey. CS Technical Report, University of Wisconsin–Madison (2009)
7. Bicocchi, N., Mamei, M., Prati, A., Cucchiara, R., Zambonelli, F.: Pervasive Self-Learning with Multi-modal Distributed Sensors. In: 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 61–66, IEEE Press, Washington (2008)
8. Frinken, V., Bunke, H.: Self-training Strategies for Handwriting Word Recognition. In: Perner P. (eds.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 5633, pp. 291–300. Springer, Heidelberg (2009)
9. Guzmán-Cabrera, R., Montes-Y-Gómez, M., Rosso, P., Villaseñor-Pineda, L.: A Web-Based Self-training Approach for Authorship Attribution. In: 6th international conference on Advances in Natural Language Processing, pp. 160–168, Springer-Verlag press (2008)
10. Feng-Biao, G., Chun-Ting, Z.: ZCURVEV: a new self-training system for recognizing protein-coding genes in viral and phage genomes. BMC Bioinformatics, 7, 9 (2006)
11. Blum, A. and Mitchell, T.: Combining labeled and unlabeled data with co-training. In: 11th annual conference on Computational learning theory, pp. 92–100, Morgan Kaufmann press (1998)
12. Guan, D., Yuan, W., Lee, Y.K., Gavrilov, A., Lee, S.: Activity Recognition Based on Semi-supervised Learning. In: IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 469–475, IEEE press, Washington, (2007)
13. Zhou, Y., Goldman, S.:Democratic Co-Learning. In: 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 594–602, IEEE press, Washington, (2004)
14. Project Confidence, `http://www.confidence-eu.org/`
15. Dovgan, E., Luštrek, M., Pogorelc, B., Gradišek, A., Burger, H., Gams, M.: Intelligent elderly-care prototype for fall and disease detection from sensor data. Zdravniski Vestnik - Slovenian Medical Journal 80(11), 824–831 (2011)
16. Kaluža, B., Mirchevska, V., Dovgan, E., Lutrek, M., Gams, M.: An Agent-Based Approach to Care in Independent Living. In: De Ruyter, B., Wichert, R., Keyson, D., Markopoulos, P., Streitz, N., Divitini, M., Georgantas, N., Mana Gomez, A. (eds.) AmI 2010. LNCS, vol. 6439, pp. 177–186. Spriner, Heidelberg (2010)
17. Ubisense, `http://www.ubisense.net`

18. Kaluža, B., Dovgan, E.: Glajenje trajektorij gibanja cloveškega telesa zajetih z radijsko tehnologijo. In: Proceedings of the 12th International Multiconference Information Society IS 2009, vol. A, 97–100, Ljubljana, Slovenia (2009)
19. Luštrek, M., Kaluža, B.: Fall detection and activity recognition with machine learning. Informatica, 33, 197–204 (2009)
20. Yanhua, C., Manjeet, R., Ming, D., Jing, H.: Non-negative matrix factorization for semi-supervised data clustering. Knowledge and Information Systems, 17, 355-379 (2008)
21. UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
22. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. MIT Press, Cambridge, MA (1999)
23. Ross, Q.J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1992)
24. Hall, M., Frank, E., Holems, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update. In: SIGKDD Explorations, 11, 10-18 (2009)
25. Breiman, L.: Random Forests. Machine Learning., 45, 5–32 (2001)
26. Gjoreski, H.: Adaptive human activity/posture recognition and fall detection using wearable sensors. Master thesis, Jožef Stefan International Postgraduate School (2011)