

Transferring Knowledge of Activity Recognition across Sensor Networks

T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse

Intelligent Systems Lab Amsterdam
University of Amsterdam
Science Park 107,1098 XG, Amsterdam
The Netherlands
T.L.M.vanKasteren@uva.nl
<http://www.science.uva.nl/~tlmkaste>

Abstract. A problem in performing activity recognition on a large scale (i.e. in many homes) is that a labelled data set needs to be recorded for each house activity recognition is performed in. This is because most models for activity recognition require labelled data to learn their parameters. In this paper we introduce a transfer learning method for activity recognition which allows the use of existing labelled data sets of various homes to learn the parameters of a model applied in a new home. We evaluate our method using three large real world data sets and show our approach achieves good classification performance in a home for which little or no labelled data is available.

1 Introduction

Automatically recognizing activities, such as cooking, sleeping and bathing, in a home setting allows many applications in areas such as intelligent environments [2,7] and healthcare [1,22,29]. It is to be foreseen that in the near future activity recognition systems will be installed on a large scale (i.e. in many homes). Most state of the art activity recognition models are supervised models that require labelled data to learn the model parameters [10,16,25,27]. Because of differences in both the layout of houses and the behaviour of their inhabitants, a model trained for one house cannot be used for another house. This means that a labelled dataset needs to be recorded for each house. Since this is expensive, we propose to use transfer learning [3,5,24] to transfer knowledge from labelled datasets to situations where no or little labelled data is available.

Transfer learning has been successfully applied to independent and identically distributed (i.i.d.) data, using discriminative models [14,21]. Activity recognition, however, presents us with two important challenges: First, our measurements are part of a time series, and are therefore not i.i.d. Second, we deal with situations where the data from a house is largely unlabelled, hence making discriminative models inadequate. In this paper, we propose a method for applying transfer learning to time series (where the data points are not independent), using a generative model to allow the use of both labelled and unlabelled data

during learning. We apply our method in the health care domain where the goal is to recognize activities of daily living (ADL) from wireless sensor network data. The list of ADLs is a well recognized fixed list of activities which are good indicators for the cognitive and physical wellbeing of elderly [13]. In our experiments we recognize the same set of ADLs in different houses, having different sensor networks. Three large real world datasets are used to evaluate the performance of our method in activity recognition.

The rest of this paper is organized as follows. Chapter 2 describes related work of both activity recognition and transfer learning. In chapter 3 we describe our transfer learning approach in detail. Chapter 4 discusses the experiments and results. Finally, in chapter 5 we sum up our conclusions.

2 Related Work

This section describes the related work of activity recognition systems and transfer learning approaches. Furthermore, the terminology is introduced which is used throughout the rest of the paper.

2.1 Activity Recognition Systems

Activity recognition systems consist of a sensing system for obtaining observations and a recognition model which interprets these observations and recognizes which activities are performed. Sensing systems may include camera's [10], RFID [19,30], wearables [11,15] and wireless sensor networks [23,27].

Several models for activity recognition have been proposed, mainly of probabilistic nature. Good results are obtained using generative models such as the hidden Markov model (HMM) [19,27] and discriminative models such as conditional random fields (CRF) [6,27]. Extensions to these models such as hierarchical models [16,18] and segment models [10,25], have been proposed to deal with the long term dependencies in activities.

All these models are supervised models and therefore require labelled data to learn the model parameters. Some models have been proposed that somewhat reduce the need for supervised data such as a hybrid generative and discriminative model [12] or models that use common-sense knowledge from the web [31]. Such models provide interesting new opportunities for modelling activity recognition. However, the advantage of our method is that any existing or upcoming generative model that has proven itself in the field of activity recognition can be used without altering the model. That is, we can simply use the proposed model and learn its parameters using our transfer learning approach.

2.2 Transfer Learning

Transfer learning refers to techniques that learn model parameters for a classification task by incorporating training data from different, but related classification tasks. We distinguish between *source* tasks that provide us with training data,

and a *target* task which is the actual classification task we are interested in. Early work on transfer learning primarily focused on multi-task learning in which several tasks were learned jointly, yielding a better performance than learning the tasks separately [3,5,24].

For example, the goal in newsgroup classification tasks is to classify which newsgroup a particular document belongs [9,21]. One task is to recognize if a document comes from a newsgroup about space or about hardware. When including training data of other newsgroups such as religion, baseball and motorcycles the performance improves significantly [21]. This is because the other newsgroups provide information about the co-occurrence of words. A word such as ‘moon’ might often occur together with the word ‘rocket’. If the word ‘rocket’ did not occur in the space newsgroup dataset but the word ‘moon’ did, the classifier can still learn that ‘rocket’ is descriptive for the space newsgroup. Because it occurs often together with ‘moon’ in the other datasets.

The optimal way to perform transfer learning is still an active topic of research. One approach that seems to work well with probabilistic models is the use of a prior distribution over the model parameters. The prior provides an initial estimate of the model parameters for target task and is learned from the source tasks [14,21]. The influence of the prior decreases as more training data is observed, therefore providing a natural mechanism to balance the effect of the prior distribution and the training data while learning the model parameters.

3 Transfer Learning for Activity Recognition

When applying transfer learning to activity recognition each classification task corresponds to a house in which we perform activity recognition. We distinguish between a target house, for which there is little or no training data available, and a number of source houses for which we have large labelled datasets. The same list of ADLs is used for each house, while the sensor network for each house is different. To perform transfer learning from the source houses to the target house, two problems need to be solved:

1. How do we deal with differences in sensor networks due to the different layout of houses?
2. How can we learn model parameters such that differences in behaviour of the inhabitants are taken into account?

The first problem involves differences in feature spaces between the houses. Because each house has a different layout, the sensor network in each house has a different configuration, resulting in a different feature space. For example, one house might have a separate room for the toilet and bathroom, while these may be built together in another. As a result, the sensors used will differ, both in number and in function. To solve this we need to introduce some kind of mapping allowing us to have a single common feature space that can be used for all houses. We use meta features [14] for this mapping, which are features that describe the properties of the actual features. Each sensor is described by one

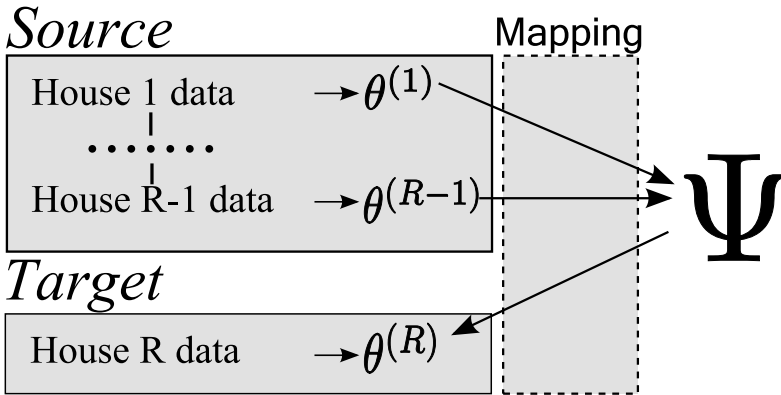


Fig. 1. Graphical representation of transfer learning framework. For each source house i training data is used to learn model parameters $\theta^{(i)}$. All the source model parameters are used to learn the hyperparameters Ψ of the prior distributions. Which in turn is used to learn the target model parameters $\theta^{(R)}$ together with any available data from that house.

or more meta features, for example, a sensor on the microwave might have one meta feature describing the sensor is located in the kitchen, and another that the sensor is attached to a heating device.

The second problem involves differences in behaviour between inhabitants. Even though for each house the same activity labels are used, there may still be differences in how activities are performed. For example, one person might often have cereal for breakfast, while another prefers toast. Such differences in behaviour require different sets of parameters to allow the model to recognize the corresponding activities. Therefore, we use a separate model for each house, each having its own set of model parameters. A prior distribution is learned from the source houses and used to provide a sensible initial value for the model parameters of the target house. Specific behaviour can then be accounted for by further updating the parameters using unlabelled and/or labelled training data from the target house. The entire approach is shown in a diagram in Figure 1.

In the rest of this section we first explain the type of mapping and the activity recognition model that we use. Then we explain how we learn the prior distribution and how this prior distribution is used to learn the parameters of the target model.

3.1 Mapping Using Meta Features

We define a sensor feature space as the original feature space in which each sensor of a house represents a feature, while the meta feature space is represented by meta features. There is a separate sensor feature space for each house, while the meta feature space shared by all houses. Choosing a proper mapping is difficult, since the optimal choice is not clear and a wrong decision can strongly

Table 1. Example of sensors (horizontal) being represented by meta features (vertical) for two houses

House	Sensors	Bathroom	Entrance	Bathroom	Other	Kitchen	Heating	Kitchen	Storage	Kitchen	Other	Outside	Entrance	Bedroom	Entrance	Bedroom	Other	Toilet
House A	Microwave	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stove	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	...																	
House B	Microwave	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Refrigerator	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	...																	

affect the performance of the model. In previous work on transfer learning for activity recognition a comparison of mappings was made [26]. The mapping that combined sensor readings in a single feature based on their function (e.g. sensors used during cooking) gave the best results. We use the same type of mapping in the form of meta features, by defining meta features that describe the function of sensors (Table 1).

It is important to notice is that we do *not* first map all the sensor data from the sensor feature space to the meta feature space. Instead, as can be seen in Figure 1, this mapping occurs when learning the prior distribution, and using the prior to learn the target model parameters. However, it is possible to first map all the sensor data to the meta feature space and then create a single model for all houses. This approach was taken in [26] and we compare the performance of our approach to that approach in the experiment section.

3.2 Model for Activity Recognition

To recognize the activities from sensor data we use the hidden Markov model (HMM), which has been shown to perform well in this domain [19,27]. The HMM is defined in terms of an observable variable \mathbf{x}_t (the features in the sensor feature space) and a hidden variable y_t (the activities to recognize) at each time slice. Two dependency assumptions specify the model,

- The hidden variable at time t , namely y_t , depends only on the previous hidden variable y_{t-1} (*Markov assumption* [20]).
- The observable variable at time t , namely \mathbf{x}_t , depends only on the hidden variable y_t at that time slice.

These assumptions allow us to factorize the joint probability over all variables as follows

$$p(\mathbf{y}_{1:T}, \mathbf{x}_{1:T}) = p(y_1) \prod_{t=1}^T p(\mathbf{x}_t | y_t) \prod_{t=2}^T p(y_t | y_{t-1}). \quad (1)$$

Table 2. Overview of the distributions used in the HMM, parameterized by the model parameters $\theta = \{\pi, A, B\}$. And the corresponding prior distribution, parameterized by the hyperparameters $\Psi = \{\eta, \rho, \omega, v\}$.

Factor	Model Distribution		Prior Distribution	
Name	Name	Parameters	Name	Hyperparameters
Initial State	Multinomial	π	Dirichlet	η
Transition	Multinomial	A	Dirichlet	ρ
Observation	Binomial	B	Beta	ω, v

The different factors represent: the initial state distribution $p(y_1)$ parameterized by π ; the observation distribution $p(\mathbf{x}_t | y_t)$ parameterized by B ; the transition distribution $p(y_t | y_{t-1})$ parameterized by A . The entire model is therefore parameterized by a set of three parameters $\theta = \{\pi, A, B\}$.

For more technical details about distributions used in the HMM we refer the reader to appendix A.

3.3 Learning the Prior Distribution

In Bayesian statistics a prior is said to be conjugate if the resulting posterior is of the same functional form as the prior [4]. The parameters of prior are typically called hyperparameters Ψ , to clearly distinguish them from the model parameters θ . We use conjugate priors in this work, an overview of all the distributions and their parameters can be found in Table 2.

To learn the hyperparameters we first learn the model parameters of the source houses. Because we have large labelled datasets for the source houses, we can easily learn those parameters using maximum likelihood. These source model parameters provide us with examples of what the model parameters look like and are used to learn the hyperparameters.

Learning the hyperparameters of the initial state distribution and the transition distributions is straightforward, because the dimensionality of the model parameters is the same for all houses. We can calculate them efficiently using numerical methods [17].

Estimating the hyperparameters of the observation distributions is more involved because of the different sensor feature spaces in each house. This is where the earlier proposed mapping comes into play. We map the learned observation model parameters of the source houses to the meta feature space. Then we use those values to learn the hyperparameters using numerical methods.

For more technical details about learning the hyperparameters we refer the reader to appendix B.

3.4 Using the Prior to Learn the Target Model Parameters

To learn the model parameters of the target house we use the EM algorithm. In the E-step any available unlabelled and/or labelled data from the target house is used to calculate the expectations. During the M-step these expectations are

used to calculate the new set of parameters, only this time the prior distribution is added to that calculation.

For the initial state distribution and the transition distribution this is straightforward, because the hyperparameters are of the same dimensionality as the model parameters. However, the observation hyperparameters are stored in the meta feature space and therefore need to be mapped to the sensor feature space of the target house. The EM algorithm is run until it converges, after which transfer learning the target model parameters is completed.

For more technical details about learning the hyperparameters we refer the reader to appendix C.

4 Experiments

We want to find a good method for transfer learning in activity recognition. Our proposed approach is characterized by three elements: 1. Meta features are used to map between the sensor feature space and to a common meta feature space; 2. A separate model is used for each house to take into account the differences in behaviour of the inhabitants and 3. A generative model is used to allow the inclusion of unlabelled data of the target house during the learning process.

To validate how well this approach works we perform the following experiments. First, we compare the performance of a model using the meta-feature space for representing the sensor data, to a model using the original sensor feature representation. Second, we compare the performance of using a separate model for each house, to the performance of a single model for all houses. Third, we compare the performance of using both labelled and unlabelled data, to the performance of using only labelled data.

We first give a description of the houses and the datasets recorded in them and provide details of our experimental setup. Then we present the results and discuss the outcome.

4.1 Sensor System

In this work we apply our method to wireless sensor network data, using both existing and novel real world datasets. Our wireless sensor network consists of several wireless network nodes to which sensors can be attached. Examples of sensors used include reed switches to measure open-close states of doors and cupboards; pressure mats to measure sitting on a couch or lying in bed; mercury contacts for movement of objects (e.g. drawers); passive infrared (PIR) to detect motion in a specific area; float sensors to measure the toilet being flushed. Sensor output is binary and represented in a feature space which is used by the model to recognize the activities performed.

4.2 Data

Our sensor system was used to record datasets in three houses. One three room apartment (house A), one two room apartment (house B) and one two story

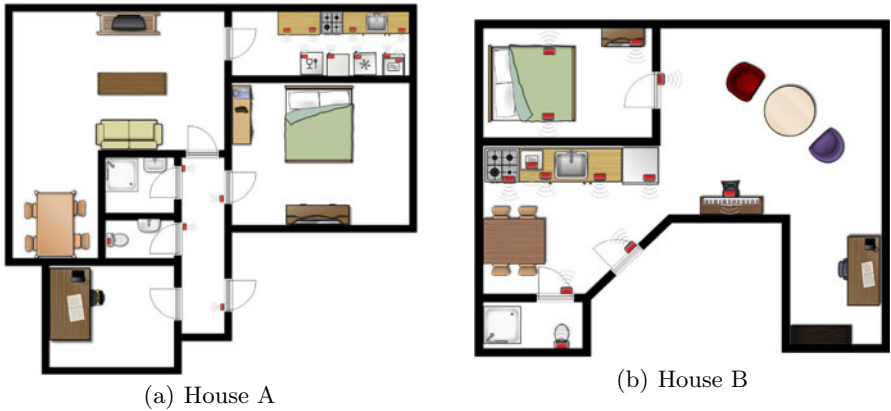


Fig. 2. Floorplan of houses A and B, the red boxes represent wireless sensor nodes. (created using: <http://www.floorplanner.com/>)

house (house C), an overview of the datasets can be found in table 3. The datasets are available at: <http://www.science.uva.nl/~tlmkaste>.

The layout of the houses differs strongly, for example, there are two toilets in house C, the toilet in house B is in the same room as the shower, while the toilet and shower in house A are in separate rooms. Furthermore, the inhabitants differ as well, house A was occupied by a 26 year old male, house B by a 28 year old male and house C by a 57 year old male. To further illustrate the differences between the houses we have included the floorplans of houses A and B (Fig. 2) and house C (Fig. 3).

We asked the inhabitants to annotate their behaviour using eight activities based on the list of activities of daily living (ADLs), a health care standard for monitoring elderly [13]. The activities in house A and B were annotated using a wireless bluetooth headset, the inhabitant recorded the start and end point of an activity while performing it. In house C activities were annotated using a handwritten diary. The activities annotated are the same for all three houses

Table 3. Information about the datasets recorded in three different homes using a wireless sensor network

House	House A	House B	House C
Age	26	28	57
Gender	Male	Male	Male
Setting	Apartment	Apartment	House
Rooms	3	2	6
Duration	25 days	13 days	18 days
Sensors	14	23	21
Recorded by	Authors	Authors	Authors

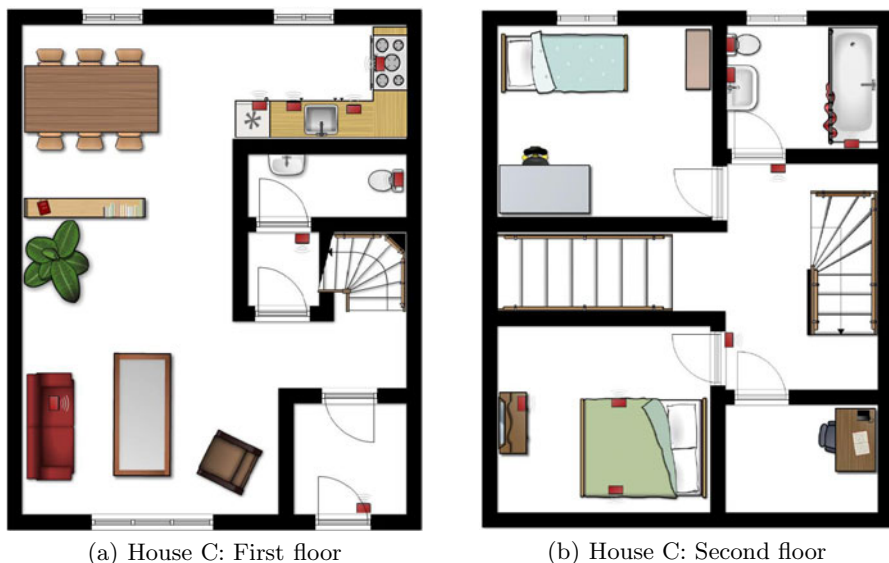


Fig. 3. Floorplan of house C, the red boxes represent wireless sensor nodes. (created using: <http://www.floorplanner.com/>)

and can be found in Table 4. Timeslices for which no annotation is available are collected in a separate activity labelled as ‘other activity’.

4.3 Experimental Setup

In all experiments the HMM was used as activity recognition model. All mappings that are performed use the meta feature list shown in Table 1, as discussed in section 3.1. Sensor data is discretized in timeslices of length $\Delta t = 60$ seconds. This time slice length is long enough to provide a discriminative sensor pattern and short enough to provide high resolution labelling results. After discretization we have a total of 35486 timeslices for house A, 17350 timeslices for house B and 26236 timeslices for house C.

We split our data into a test and training set using a ‘leave one day out’ approach. In this approach, one full day of sensor readings is used for testing and the remaining days are, depending on the experiment, either partly or fully used for training. We use each day as a test day once and report the average of the performance measure.

We evaluate the performance of our models using the F-measure, which is calculated from the precision and recall scores. We are dealing with a multi-class classification problem and therefore define the notions of true positive (TP), false negatives (FN) and false positives (FP) for each class separately as shown in a confusion matrix in Table 5, where N is the total number of classes.

Table 4. The activities that were annotated in the different houses. The ‘Num.’ column shows the number of times the activity occurs in the dataset. The ‘Time’ column shows the percentage of time the activity takes up in the dataset. All unannotated timeslices were collected in a single ‘Other’ activity.

Activity	House A		House B		House C	
	Num.	Time	Num.	Time	Num.	Time
Leave house	33	50.5%	16	50.6%	47	45.7%
Toileting	114	1.0%	28	0.6%	89	1.0%
Take shower	23	0.8%	8	0.6%	14	0.8%
Brush teeth	16	0.1%	12	0.2%	26	0.4%
Go to bed	24	33.2%	11	30.7%	19	29.2%
Prepare Breakfast	20	0.3%	7	0.5%	18	0.6%
Prepare Dinner	9	0.9%	2	0.2%	11	1.1%
Get drink	20	0.2%	11	0.2%	10	0.1%
Other	-	13.0%	-	16.4%	-	21.1%

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (2)$$

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i} \quad (3)$$

$$\text{F-Measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

4.4 Experiment 1: Meta-features vs. Sensor Feature Space

In this experiment we compare the performance of using the original sensor feature space to the performance of using the meta feature representation. We do not use any form of transfer learning in this experiment because it is not possible to do transfer learning using the original sensor feature space. Instead we train the model parameters using only data from a single house by performing maximum likelihood estimation, so no prior is used in learning the parameters.

Table 5. Confusion Matrix showing the true positives (TP), false negatives (FN) and false positives (FP) for each class. The ϵ_{ij} terms show the error between true class i and inferred class j . FN is the sum of the error terms in the same row, FP the sum of the error terms in the same column.

True	Inferred			FN
	1	2	3	
1	TP_1	ϵ_{12}	ϵ_{13}	FN_1
2	ϵ_{21}	TP_2	ϵ_{23}	FN_2
3	ϵ_{31}	ϵ_{32}	TP_3	FN_3
FP	FP_1	FP_2	FP_3	Total

In the case of the meta feature space the sensor data is mapped to the meta feature space. For the sensor feature space the sensor data can be used as it is. This allows us to do a fair comparison of the two feature spaces.

Figure 4 shows the results for all three houses. The X-axis shows the number of days of labelled data that was used, any remaining unlabelled data was also included during learning. The plot shows that the performance of the model using the meta feature space is slightly less than the performance of the model using the sensor feature space. This is because several sensors are combined into a single meta feature which gives the model less information for distinguishing activities.

4.5 Experiment 2: Separate Model vs. Single Model vs. No Transfer Learning

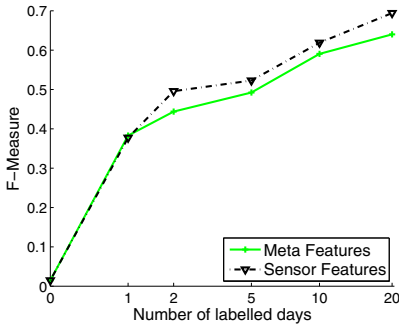
This experiment compares our transfer learning approach in which a separate model is used for each house with a transfer learning approach which uses a single model for all houses. A single house is used as target house, while the remaining two houses are used as source house. We compare the performance of these two transfer learning approaches to the performance of the model from the previous experiment in which no transfer learning and no mapping was used. This way we are able to see which transfer learning method works best and what the difference in performance is compared to not doing transfer learning.

Figure 5 shows the results for all three target houses. The X-axis show the number of days of labelled data that was used, any remaining unlabelled data was also included during learning. First of all we see that both our approach and the single model approach strongly outperform the ‘no transfer learning’ approach in all three houses when 0 days of labelled training data are used. This is because the ‘no transfer learning’ approach has no labelled data to learn its parameters, while the two transfer learning approaches can use the labelled data of the source houses.

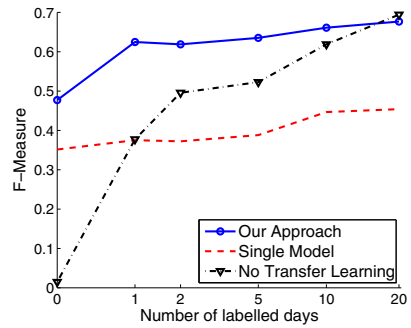
Furthermore, we see that our approach strongly outperforms the ‘single model’ approach in case of house A. This shows the benefit of having a separate model for each house, as can be seen from the jump in performance of going from 0 days of labelled data to 1 day of labelled data. Our approach is able to learn model parameters that take into account the specific behaviour for that house. The ‘single model’ on the other hand only gains a slight performance increase from this extra data, because it still shares the labelled data with the labelled data from the source houses. This makes the weight of the labelled data of the target house much less than when a prior is used.

Our approach also outperforms the ‘no transfer learning’ approach, although the difference in performance decreases as the number of labelled days increases. This clearly shows how the use of a prior helps in learning the model parameters and that its effect decreases as more labelled data is used.

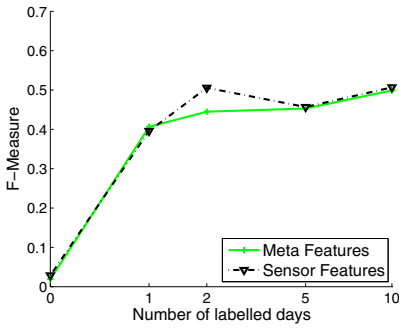
In the house B plot we see the ‘no transfer learning’ approach sometimes outperforms the transfer learning approach. This phenomenon is called negative transfer [5] which means that sometimes transfer learning can have a negative effect on the learning process. The reason for this is that it is not clearly defined



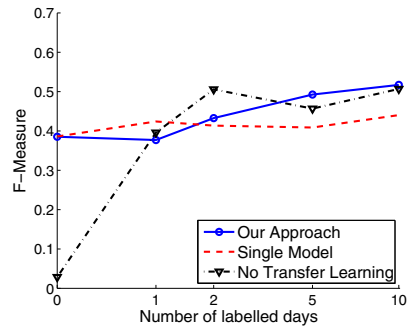
(a) House A



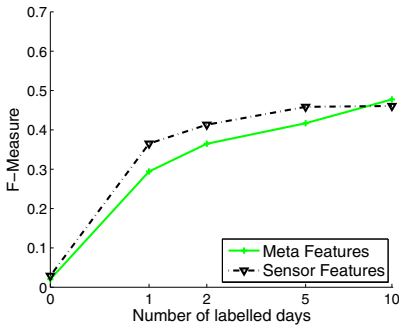
(a) House A



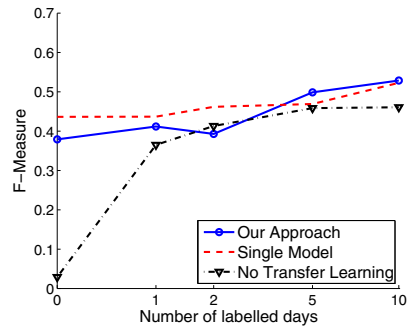
(b) House B



(b) House B



(c) House C



(c) House C

Fig. 4. Results of experiment 1, comparing the performance of the HMM using the meta-feature space and the original feature space. The x-axis are in log-scale and show the number of labelled days of data that were used for training.

Fig. 5. Results of experiment 2, comparing the performance of our transfer learning approach using a separate model for each house with a transfer learning approach using a single model for all houses and with an approach where no transfer learning is used. The x-axis are in log-scale and show the number of labelled days of data that were used for training.

which parts of data in the source houses are useful for the target house and which or not. Including training data from the source houses during learning can therefore sometimes pull the choice of parameters away from the optimal solution.

The house B plot also shows that our approach does slightly worse than the ‘single model’ approach when using 1 day of labelled data, but does better when more days of labelled data are included. This shows the advantage of using a prior, as more target data becomes available the learning method has to rely less on the prior (which caused the negative transfer). On the other hand, in the single model approach the data from both the source and target houses are all considered as valid training data for the single model. Therefore, a lot more target house data needs to be observed before it can outweigh the source house data, which is causing the negative transfer.

Finally, in the house C plot we see the ‘single model’ outperforms our approach when few days of labelled data are used, but as more days are added our approach manages to perform better or equal. This is similar to what we observed in the house B plot. Our approach slightly outperforms the ‘no transfer learning’ approach when a large number of labelled days is used.

4.6 Experiment 3: Labelled vs. Unlabelled Data

The use of generative models allows us to include unlabelled data during the learning process. In this experiment we compare performance of using both unlabelled and labelled data to using only labelled data. In both cases we use our transfer learning approach to learn the model parameters. The results for the various houses can be found in Figure 6.

We see that adding unlabelled data increases performance for house A, gives more or less equal performance for house B and decreases performance for house C, compared to the labelled only approach. The explanation of these mixed results is that the success of adding unlabelled data depends on the quality of the labelled data. We suspect that the use of a hand written diary for annotation (used in house C) results in less accurate annotation than using the bluetooth headset method (used in houses A and B). Although this less accurate annotation does not affect the learning process when using unlabelled data. It does affect the validation process when verifying if the inferred labels are correct.

4.7 Discussion

The results of our experiments show that our transfer learning method works well. Especially when no labelled data is available for a target house, our transfer learning approach is able to provide a good estimate of the parameters. But also in the presence of labelled data it can help in learning the model parameters. In some cases negative transfer can result in a lower performance compared to a non-transfer learning approach. This phenomenon has been reported in other transfer learning scenarios as well, but it is not well understood how to solve it [5].

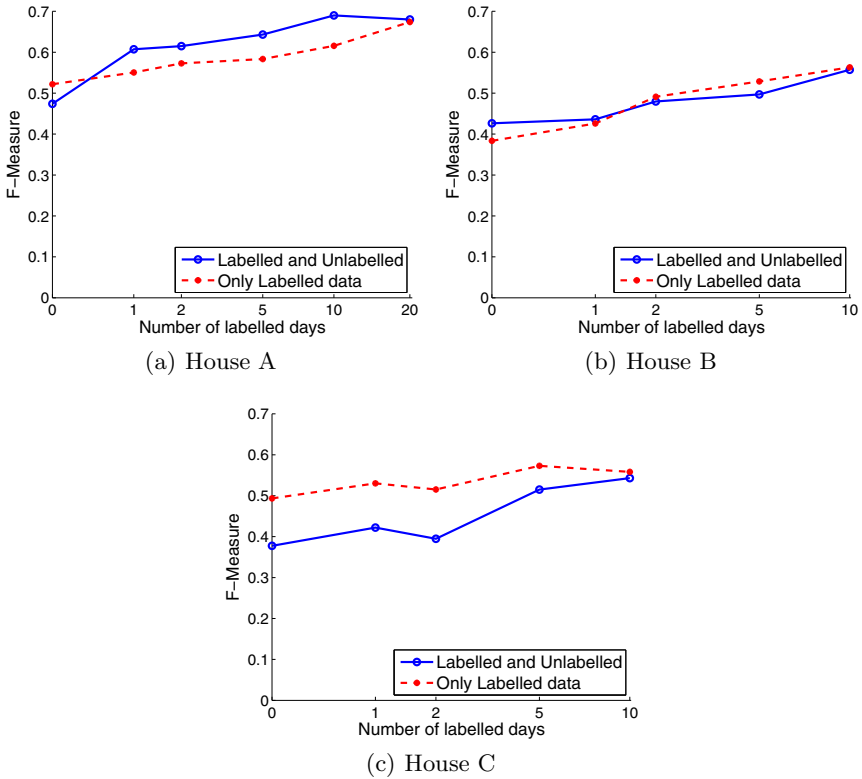


Fig. 6. Results of experiment 3, comparing the performance of using only labelled data with using both labelled and unlabelled data. The x-axis are in log-scale and show the number of labelled days of data that were used for training. In the case where unlabelled data is also included, the remaining days of unlabelled data are added during training.

An important factor in successfully applying transfer learning is the use of a proper mapping. In this work we manually defined the mapping beforehand. An alternative is to learn the mapping automatically from data [8,32]. However, because we are working with time series data trying various kinds of mappings might result in too high computation times for the approach to be feasible.

In terms of future work, it would be interesting to apply transfer learning to several other models. For example, the use of hierarchical models might be better fit for transfer learning because the different levels of the hierarchy allow a better abstraction between houses. Comparing the performance gain due to transfer learning between several models can provide interesting insights on how to accurately model data. It would also be interesting to apply our transfer learning approach to other sensing modalities such as camera's or wearables. Creating a proper mapping for those modalities will be challenging. Finally, it would be interesting to perform transfer learning across different sensing modalities. For example, using source houses in which camera's and wearables are used

to perform activity recognition and a target house in which a wireless sensor network is used.

5 Conclusion

We have addressed the problem of learning model parameters when little or no labelled data is available for the house activity recognition is to be performed in. Our main contribution is the introduction of a transfer learning method for activity recognition, which uses a prior to transfer general knowledge about activity recognition and allows the use of labelled and unlabelled data to learn house-specific behaviour.

Using experiments on three large real world datasets we showed our method gives good performance in activity recognition for a house for which little or no labelled data is available. The method can outperform a model trained using conventional maximum likelihood estimation. Furthermore, it can outperform a previously introduced transfer learning method in which a single model is used for all houses.

Acknowledgment

This work is part of the Context Awareness in Residence for Elders (CARE) project and the ‘Zorg(en) voor Morgen’ project. The CARE project is funded by the Centre for Intelligent Observation Systems (CIOS) which is a collaboration between UvA and TNO. The ‘Zorg(en) voor Morgen’ project is funded through the Pieken in de Delta-program by the Ministry of Economic Affairs and the cities of Utrecht and Lelystad and the provinces of Utrecht, Noord-Holland and Flevoland.

References

1. Abowd, G., Bobick, A., Essa, I., Mynatt, E., Rogers, W.: The aware home: Developing technologies for successful aging. In: Proceedings of AAAI Workshop and Automation as a Care Giver (2002)
2. Augusto, J.C., Nugent, C.D. (eds.): Designing Smart Homes, The Role of Artificial Intelligence. LNCS, vol. 4008. Springer, Heidelberg (2006)
3. Baxter, J.: A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning* 28(1), 7–39 (1997)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Heidelberg (2006)
5. Caruana, R.: Multitask learning. In: *Machine Learning*, pp. 41–75 (1997)
6. Chieu, H.L., Lee, W.S., Kaelbling, L.P.: Activity recognition from physiological data using conditional random fields. In: SMA Symposium. MIT Alliance, Singapore (2006)
7. Cook, D.J., Das, S.K.: Smart Environments: Technology, Protocols and Applications. Wiley-Interscience, Hoboken (2004)

8. Dai, W., Chen, Y., Xue, G.-R., Yang, Q., Yu, Y.: Translated learning: Transfer learning across different feature spaces. In: NIPS 2008: Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS 2008), Vancouver, Canada (2008)
9. Dai, W., Xue, G.-R., Yang, Q., Yu, Y.: Transferring naive bayes classifiers for text classification. In: AAAI, pp. 540–545 (2007)
10. Duong, T., Phung, D., Bui, H., Venkatesh, S.: Efficient duration and hierarchical modeling for human activity recognition. *Artif. Intell.* 173(7-8), 830–856 (2009)
11. Huynh, T., Schiele, B.: Towards less supervision in activity recognition from wearable sensors. In: Proceedings of the 10th IEEE International Symposium on Wearable Computing (ISWC), Montreux, Switzerland (October 2006)
12. Huynh, T., Schiele, B.: Towards less supervision in activity recognition from wearable sensors. In: ISWC, pp. 3–10 (2006)
13. Katz, S.: Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. *J. Am. Geriatrics Soc.* 31(12), 721–726 (1983)
14. Lee, S.-I., Chatalbashev, V., Vickrey, D., Koller, D.: Learning a meta-level prior for feature relevance from multiple related tasks. In: ICML 2007: Proceedings of the 24th international conference on Machine learning, pp. 489–496. ACM, New York (2007)
15. Lester, J., Choudhury, T., Kern, N., Borriello, G., Hannaford, B.: A hybrid discriminative/generative approach for modeling human activities. In: IJCAI, pp. 766–772 (2005)
16. Liao, L., Fox, D., Kautz, H.: Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research* 26(1), 119–134 (2007)
17. Minka, T.P.: Estimating a dirichlet distribution. Technical report, Microsoft Research (2000)
18. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.* 96(2), 163–180 (2004)
19. Patterson, D.J., Fox, D., Kautz, H.A., Philipose, M.: Fine-grained activity recognition by aggregating abstract object usage. In: ISWC, pp. 44–51. IEEE Computer Society, Los Alamitos (2005)
20. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2), 257–286 (1989)
21. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: ICML 2006: Proceedings of the 23rd international conference on Machine learning, pp. 713–720. ACM Press, New York (2006)
22. Suzuki, R., Ogawa, M., Otake, S., Izutsu, T., Tobimatsu, Y., Izumi, S.-I., Iwaya, T.: Analysis of activities of daily living in elderly people living alone. *Telemedicine* 10, 260 (2004)
23. Tapia, E.M., Intille, S.S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha, A., Mattern, F. (eds.) PERSVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
24. Thrun, S.: Is learning the nth thing any easier than learning the first? *Advances in Neural Information Processing Systems* 8, 640–646 (1996)

25. Truyen, T.T., Phung, D.Q., Bui, H.H., Venkatesh, S.: Hierarchical semi-markov conditional random fields for recursive sequential data. In: Neural Information Processing Systems, NIPS (2008)
26. van Kasteren, T., Englebienne, G., Kröse, B.: Recognizing activities in multiple contexts using transfer learning. In: Proceedings of the AAAI Fall Symposium on AI in Eldercare: New Solutions to Old Problems. AAAI Press, Menlo Park (2008), ISBN=978-1-57735-394-2
27. van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: UbiComp 2008: Proceedings of the 10th international conference on Ubiquitous computing, pp. 1–9. ACM, New York (2008)
28. Williams, M.T.: Beta-binomial distribution for proportional confidence intervals. Technical report, University of Leeds (1998)
29. Wilson, D.H.: Assistive Intelligent Environments for Automatic Health Monitoring. PhD thesis, Carnegie Mellon University (2005)
30. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: ICCV, pp. 1–8 (2007)
31. Wyatt, D., Philipose, M., Choudhury, T.: Unsupervised activity recognition using automatically mined common sense. AAAI, 21–27 (2005)
32. Zhang, J., Ghahramani, Z., Yang, Y.: Learning multiple related tasks using latent independent component analysis. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18, pp. 1585–1592. MIT Press, Cambridge (2006)

A Probability Distributions Used in the Hidden Markov Model

The HMM factorizes the joint probability over the observations and activities as

$$p(y_{1:T}, \mathbf{x}_{1:T}) = p(y_1) \prod_{t=2}^T p(y_t | y_{t-1}) \prod_{t=1}^T p(\mathbf{x}_t | y_t). \quad (5)$$

The individual factors are distributed as

$$p(y_1) \equiv \prod_{i=1}^K \pi_i^{\delta(y_1-i)} \quad (6)$$

$$p(y_t | y_{t-1} = i) \equiv \prod_{j=1}^K a_{ij}^{\delta(y_t-j)} \quad (7)$$

$$p(\mathbf{x}_t | y_t) = \prod_{n=1}^N p(x_t^n | y_t) \quad (8)$$

$$(x_t^n = v | y_t = i) = (\mu_{in})^v (1 - \mu_{in})^{1-v} \quad (9)$$

where $\delta(x)$ is the dirac delta function giving 1 if $x = 0$ and 0 otherwise.

We use conjugate priors for the model distributions of the HMM.

$$\text{Dir}(\pi \mid \eta) = \frac{\Gamma(\sum_{k=1}^K \eta_k)}{\Gamma(\eta_1) \dots \Gamma(\eta_K)} \prod_{k=1}^K \pi_k^{\eta_k - 1} \tag{10}$$

$$\text{Dir}(a_i \mid \rho) = \frac{\Gamma(\sum_{k=1}^K \rho_k)}{\Gamma(\rho_1) \dots \Gamma(\rho_K)} \prod_{k=1}^K a_{ik}^{\rho_k - 1} \tag{11}$$

$$\text{Beta}(\mu_{in} \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu_{in}^{\alpha - 1} (1 - \mu_{in})^{\beta - 1}. \tag{12}$$

where $\Gamma(\cdot)$ is the gamma function. The parameters α and β are further parameterized as $\alpha_{in} = v_i^T f_n$ and $\beta_{in} = \omega_i^T f_n$, where f_n is a row vector of binary meta-features as shown in Figure 1. The hyperparameters v and ω are positioned in the meta feature space.

An overview of the probability distributions and their parameters is given in Table 2.

B Estimating the Hyperparameters

The maximum likelihood estimates of the parameters of the prior distributions cannot be found in closed form. We use numerical methods for estimating these parameters [17,28]. This gives us the values of α and β which are needed to find the values of the meta feature parameters v and ω . Because f_n is given we can find the least square solution to v and ω by solving the system of equations as defined by $\alpha_{in} = v_i^T f_n$ and $\beta_{in} = \omega_i^T f_n$. To guarantee a non-negative value we add a ‘bias’ meta-feature with a large enough positive value.

C Learning the Model Parameters Using the Prior

The MAP estimates of the HMM parameters can be found in closed form solutions by taking the derivative of the expectation function with respect to the parameter of interest. By using lagrange multipliers we can constrain the maximization to satisfy the rules of probability.

$$\pi_i = \frac{p(y_1 = i \mid x_{1:T}, \theta_{old}) + (\eta_i - 1)}{\sum_{i \in y_1} \{p(y_1 = i \mid x_{1:T}, \theta_{old}) + (\eta_i - 1)\}} \tag{13}$$

$$a_{ij} = \frac{\sum_{t=2}^T p(y_t = j, y_{t-1} = i \mid x_{1:T}, \theta_{old}) + (\rho_{ij} - 1)}{\sum_{t=2}^T \sum_{j \in y_t} p(y_t = j, y_{t-1} = i \mid x_{1:T}, \theta_{old}) + (\rho_{ij} - 1)} \tag{14}$$

$$\mu_{in} = \frac{(\alpha_{in} - 1) + \sum_{t=1}^T \xi_{inv} v_t}{(\alpha_{in} + \beta_{in} - 2) + \sum_{t=1}^T \xi_{inv}} \tag{15}$$

where $\alpha_{in} = v_i^T f_n$ and $\beta_{in} = \omega_i^T f_n$.