Domain Selection and Adaptation in Smart Homes

Parisa Rashidi and Diane J. Cook

[1] Washington State University prashidi@wsu.edu
[2] Washington State University cook@eecs.wsu.edu

Abstract. Recently researchers have proposed activity recognition methods based on adapting activity knowledge obtained in previous spaces to a new space. Adapting activity knowledge allows us to quickly recognize activities in a new space using only small amounts of unlabeled data. However, adapting from dissimilar spaces not only does not help the recognition task, but might also lead to degraded recognition accuracy. We propose a method for automatically selecting the most promising source spaces among a number of available spaces. Our approach leads to a scalable and quick solution in real world, while minimizing the negative effects of adapting from dissimilar sources. To evaluate our algorithms, we tested our algorithms on eight real smart home datasets.

1 INTRODUCTION

In the past decade various supervised methods have been proposed for activity recognition from sensor data [6, 12, 15]. A problem with supervised activity recognition methods is the need for labeled data, whereas labeling human activity from sensor data is very laborious, time consuming and error-prone. Unsupervised methods such as frequent pattern mining methods [5, 16, 20] are more autonomous and less invasive, however they require large amounts of data to be collected beforehand. Therefore, the deployment process will be delayed substantially, resulting in an impractical solution in real world.

To solve this problem, domain adaptation solutions can be adopted [3, 4, 13]. Domain adaptation essentially tries to use an algorithm that has been trained on one or several source domains and adapts it to a target domain. Therefore, it is possible to skip the labeling process in a new environment and only collect a small amount of unlabeled data. A number of methods have been proposed for single source domain adaptation in smart homes. These methods include adapting learned activities for different residents [17], mapping different types of activities to each other [21], and adapting activities performed in two different spaces [7]. Recently Rashidi et al. [18] proposed a method for transferring activity knowledge from multiple spaces. They show how using multiple source domains improves the generalization capability of the algorithm and increases the chance that an activity is correctly recognized. They assume that all the sources are equally similar. In reality, this assumption can be violated, as some spaces might show a higher degree of similarity and therefore might be more suitable for domain adaptation, while some other spaces can be quite dissimilar.

Here, we explore a method of measuring the distance between a source and a target environment, called $\mathcal{H}$ distance. Using such a measure, we propose a method for selecting the most promising sources from a collection of available sources. This allows us to avoid the negative transfer effect [19]. Negative transfer effect refers to a case where adapting from a source domain might indeed result in performance degradation. It happens if the source and target domains not very similar. To the best of our knowledge, this is the first work addressing activity recognition as a multi source domain selection and adaptation problem by measuring the distance between activities performed in different environments.

2 PROPOSED MODEL

Our input data is a sequence of sensor events. Each sensor event has a timestamp and is associated with a sensor ID. In case of annotated data, each sensor event is also associated with an activity label l such as "cooking". The source domains are those smart homes in which we previously have collected data and annotated it. The target domain is a new smart home in which we intend to deploy an activity recognition system, and in which we only have collected a few days worth of unlabeled data. Note that it is possible that a hypothesis from a particular source domain might not work well on the target domain, causing the negative transfer effect. Our objective is to appropriately combine hypotheses from source domains to infer the label of activities in the target domain.

To do this, we calculate the pairwise similarity between each source and target using $\mathcal{H}$ distance [8]. It computes the distribution difference based on finite samples of data. We select the most promising sources based on the average similarity and combine them in a weighted manner.

2.1 Measuring Domain Distance

In order to be able to select the most promising domains among a list of available domains, we need to quantify the difference between the source and target environments. Selecting the most promising sources instead of using all the sources in a brute force manner allows us to avoid the negative transfer effect.

There are a number of measures for computing the distance between two distributions, such as JSD divergence [10] and KL divergence [9]. Both require distribution estimation, where in a distribution-free setting such as in our scenario, accurate estimates cannot be obtained from finite samples. Other distance measures such as L^1 or L^p norms are either too sensitive or too insensitive [8].

We use a classifier induced divergence measure called $\mathcal{H}$ distance [8] which measures the divergence only affecting the classification accuracy. Unlike other distribution distance measures such as JSD or KL divergence, it's possible to compute it from finite unlabeled samples of two distributions. More importantly, one can use the well known machine learning classification methods to estimate the distance. Blitzer et al. [3] used a variant of $\mathcal{H}$ distance as a criterion for adaptability between unlabeled domains. Here, we use it as a distance measure in order to combine hypotheses from various domains, and to discard less promising sources to avoid the negative transfer effect.

Let $\mathcal{A}$ be a family of subsets of $\Re^k$ corresponding to the characteristic functions of classifiers. Then the $\mathcal{H}$ distance between two probability distributions $\mathcal{D}$ and $\acute{\mathcal{D}}$ is defined as in Equation 1. Here sup refers to supremum.

$$d_{\mathcal{H}}(\mathcal{D}, \acute{\mathcal{D}}) = 2 * \sup_{A \in \mathcal{A}} |Pr_{\mathcal{D}}(A) - Pr_{\acute{\mathcal{D}}}(A)| \tag{1}$$

It can be shown that computing $\mathcal{H}$ distance for a finite sample is exactly the problem of minimizing the empirical risk of a classifier discriminating between instances drawn from $\mathcal{D}$ and $\acute{\mathcal{D}}$ [2]. In other words, we should label all the data points in the source domain as 0, and all the data points in the target domain as 1, regardless of their actual class labels. Then we can train a classifier to discriminate between the two domains, separating data points with a label of 0 from those data points with a label of 1. The more similar are the data points in the source and target domains, the more difficult it would be to separate the source and target data points, and therefore the higher would be the empirical risk. We re-write Equation 1 as Equation 2. Here ϵ_h is the empirical error when discriminating between source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$ based on some hypothesis $h \in \mathcal{H}_s$, and $\mathcal{H}_s$ is the hypothesis space of source domain.

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 * (1 - \arg \min_{h_j \in \mathcal{H}_s} (\epsilon_{h_j})) \tag{2}$$

We compute such a distance measure for each pair of source and target domains, and for each activity. This allows us to obtain a clear picture of domain similarity based on various activities.

2.2 Model Details

In order to facilitate domain adaptation, we have to transform the original raw features into new features. The sensor IDs are unique to each specific environment and might carry different meanings in each environment. For example, id_1 in one home might refer to an infrared motion sensor in the bedroom for detecting sleep activity. The same sensor ID id_1 in another home might refer to a cabinet sensor in the kitchen. We use the simple mapping $\mathcal{S} \rightarrow L(\mathcal{S})$ where $\mathcal{S}$ refers to the set of sensor IDs, and L refers to the location where a sensor is located (e.g. Bedroom). We maintain a unified list of locations across different environments. We use an N-gram model to represent the sequential nature of data [11]. An N-gram model for our sensor data shows the currently activated sensor, in addition to $N-1$ previously activated sensors, providing history context. We also use the start time of the activations to provide temporal context.

The source domains $\mathcal{D}_1$ through $\mathcal{D}_m$ provide us with m hypotheses h_1 .. h_m. First we select $n \leq m$ hypotheses based on the overall similarity between our target and each one of the sources. We then combine n hypotheses to infer a hypothesis h_t for the target domain. To combine hypotheses h_1 .. h_n, we weight the hypotheses by their corresponding pairwise activity similarities with the target domain, multiplied by the hypothesis confidence. Here we take the smoothness assumption [1], which states that two nearby (similar) patterns in a high density area (high confidence) should share similar decision values. Therefore we can assign a certain label to a particular activity in the target domain if the two domains are similar in terms of that particular activity, and also if our confidence is high about the predicted label of that activity. We assume that the target function h_t shares values with similar domains.

In order to use $\mathcal{H}$ distance in our activity recognition system, we need to adapt it to our scenario. Not only is it important to measure the overall similarity between source and target environments, but also it is important to know how many similar activities exist and what is their individual similarity value. We denote the distance between two domains $\mathcal{D}_s$ and $\mathcal{D}_t$ based on activity y as $d^y_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t)$. It is similar to computing the generic H distance, with the exception all the data points with a label other than y will be ignored during calculation. Note that such a distance measure is not symmetric. Consequently the activity based similarity measure is defined as in Equation 3.

$$\Phi^y(s,t) = 1 - d^y_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) \tag{3}$$

To determine similarity between a source and target domain based on individual activities, we need to identify the target activities. As we assume that the target data is unlabeled, we have no way of telling which activity a certain sensor event belongs to. To solve this problem, we utilize a two-stage algorithm. First, we build an ensemble classifier using source datasets to assign a label to the target data points. Next, we will compute the individual activity based source-target similarity, and will select the promising sources. Then we will adjust the target points labels according to the source-target activity similarity. After finding the individual activity similarities $\Phi^y(s,t)$, the overall similarity between a source and target is defined as the average similarity over all activities.

In the next step, we select the most promising sources. To select the top n sources, given n, we choose n sources with the highest similarity value $\Phi(s,t)$. If n is not given, then we need to find a suitable value for n. To select the most promising sources, we consider the overall source-target similarity $\Phi(s,t)$, in addition to the total number of selected sources. This is reflected in Equation 4. The sources with Ψ above a threshold θ_Ψ will be selected.

$$\Psi(s_i) = \frac{\Phi(\mathcal{D}_{s_i}, \mathcal{D}_t)}{\sum\limits_{s=1}^{m} \Phi(\mathcal{D}_s, \mathcal{D}_t)} * \sqrt{1 - \frac{n}{m}} \tag{4}$$

The first term in Equation 4 limits our choice of source domains to those with a similarity Φ above the average similarity. The second terms imposes a restriction on the number of selected source domains. This allows for a more efficient method, while achieving the desired accuracy.

Next, we combine the hypotheses from n selected sources. We assume that the confidence of a classifier i for a predicted label y is given by $h_i(y|x)$. Therefore, we can write the hypothesis combination rule as in Equation 5.

$$h_t(x) = \sum_{i=1}^{n} \arg\max_{y} h_i(y|x) * \Phi^y(s_i, t) \tag{5}$$

3 EXPERIMENTS

We tested our algorithms on 8 datasets collected from 6 smart homes. We refer to those 8 datasets as B1, B2, B3, K1, K2, M, T1 and T2. Apartments B1, B2, B3 were located in the same building complex. Datasets K1, K2 were collected from the same apartment, but within different time periods, housing different pairs of residents, and were annotated with different activities. The same is true for T1 and T2. All datasets are collected during a normal day to day life of residents. The list of locations included bathroom, shower, bedroom, kitchen, living room, work area, med cabinet, and entrance. A more detailed description of datasets can be found in Table 1. The sensors in our smart homes consist of infrared motion sensors, oven sensors, switch contacts on doors and cabinets, and light sensors.

Table 1: Characteristics of each dataset.

Dataset	**B1**	**B2**	**B3**	**K1**	**K2**	**M**	**T1**	**T2**
Residents	1	1	1	2	3	1+ pet	2	2
Rooms	1	1	2	3	3	3	2	2
Sensors	32	32	32	71	72	32	20	20
Activity Instances	5714	4320	3361	497	844	1513	1431	166
Collection Days	126	234	177	61	38	88	120	10
Absent Activities	W	W	–	L,R,T,E,BT	T	T,B,E	H,T,S,B,BT,W	T

To evaluate the results of our algorithms based on a ground truth, we annotated the datasets with 10 activities of interest, including hygiene(H), leaving home(L), cooking(C), relaxing(R), taking medications(T), eating(E), sleeping(S), bathing(B), bed to toilet transition(BT) and working(W). For brevity we will refer to those activities using their abbreviated form (e.g. hygiene as H). As one would expect in a real world setting, not all activities are common between different homes, nor their distributions are the same.

To test our algorithm, we considered 8 different problem settings. In each setting, 7 datasets were considered as source, and the remaining dataset as target. The base classifier is a kernel based naive Bayes classifier with a Gaussian kernel

[14]. After running our algorithm in a 10 fold cross validation manner, we found the midpoint threshold $\theta_{\Psi} = 0.5$ to be a suitable value for our experiments. The cross validation results also found the 3-Gram model to best represent the data, as higher gram values did not significantly change our results. The target and sample source datasets included 3 days of unlabeled data.

Figure 1 shows the overall activity recognition rates based on choosing the top n sources. We also performed experiments based on randomly selecting the top n sources (averaged over 10 runs), using a simple linear combination rule (no weights). One can clearly see that our domain selection algorithm outperforms random selection. It should be noted that our algorithm are based on using only a small sample of source and target datasets and it is possible that the chosen samples are not representative of the entire dataset, leading to lower recognition rates in some cases. Still, despite the fact that we are relying only on a small unlabeled target dataset, and despite the fact that the apartment layouts, sensors and residents are different, we were still able to achieve very reasonable recognition rates. In Figure 1 one can also see the effect of negative transfer. We can see that adding more data sources does not necessarily increase the recognition accuracy in the target environment. We used our source selection algorithm for choosing the best number of sources. Using our method, 95% of the maximum achievable accuracy was achieved using only 4.5 sources in average. The average accuracy was 74.88%. This shows how our algorithm can approximate the best number of promising sources, balancing efficiency and accuracy.

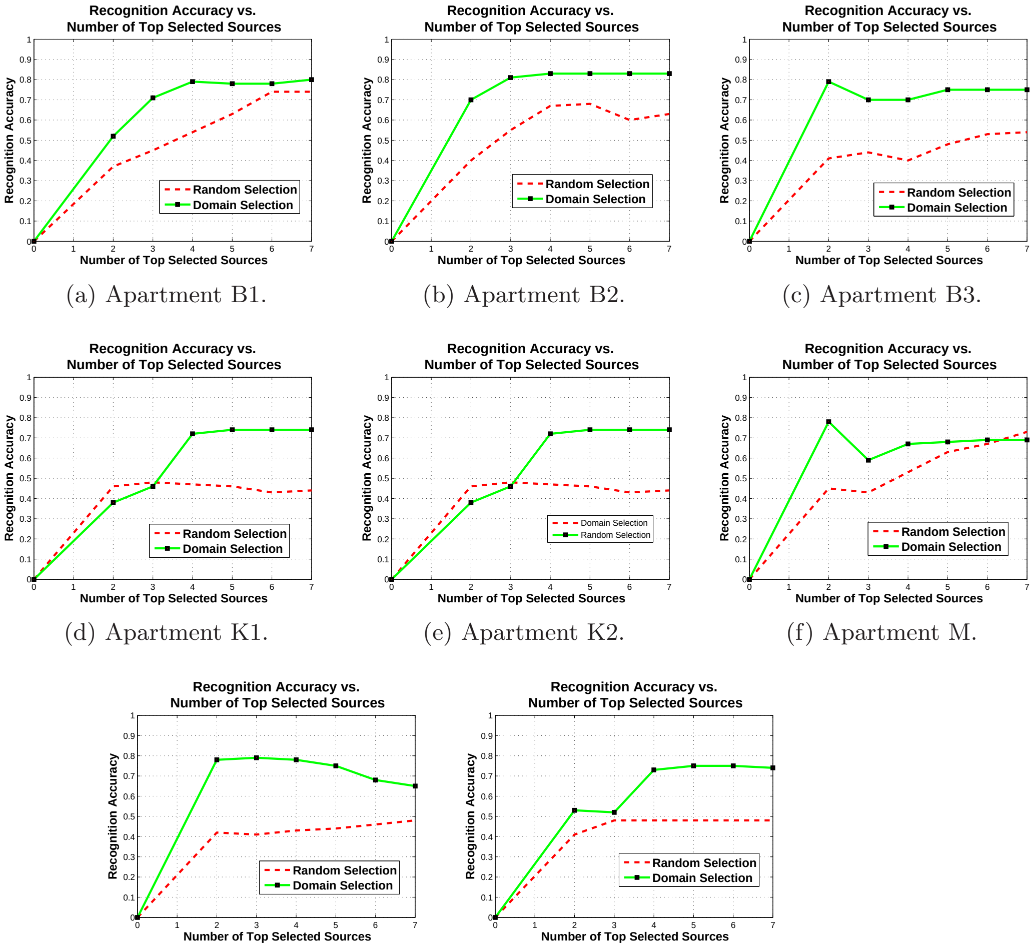
The detailed accuracies based on choosing the best number of sources are shown in Table 2. Table 2 shows accuracy results for a similar supervised method that uses all of the "labeled" target dataset for training. In contrast our method uses about 8.7% of the target dataset in average. It can be seen that though our method uses only a small fraction of "unlabeled" target dataset, it surprisingly works well. In some cases such as for the K1 dataset our algorithm even outperforms the supervised method. These experiments show how our method can be useful in a real world situation where supervised methods are expensive and impractical due to the time constraints and annotation costs.

Table 2: Comparison of pure supervised method and our method.

Dataset	**B1**	**B2**	**B3**	**K1**	**K2**	**M**	**T1**	**T2**	**Avg.**
Recognition rate: supervised.	0.86	0.86	0.9	0.63	0.79	0.73	0.88	0.76	0.80
Recognition rate: our method	0.79	0.83	0.75	0.69	0.74	0.68	0.78	0.73	0.75

4 CONCLUSIONS AND FUTURE WORK

By selecting the best source domains, we showed that it is possible to recognize activities in a target environment despite using only limited unlabeled target



(a) Apartment B1. (b) Apartment B2. (c) Apartment B3.

(d) Apartment K1. (e) Apartment K2. (f) Apartment M.

(g) Apartment T1. (h) Apartment T2.

Fig. 1: Activity recognition accuracies based on using different number of source domains.

data, and despite the fact that the residents, the number of activities, the sensor layouts and the building floorplans can be different in source and target environments. In the future, we intend to combine this method with active learning methods in order to boost our bootstrap method.

References

1. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learn-

ing Research 7, 2399–2434 (2006)

2. Ben-david, S., et al.: Analysis of representations for domain adaptation. In: Neural Information Processing Systems. MIT Press (2007)
3. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Empirical Methods in Natural Language Processing. pp. 120–128 (2006)
4. Duan, L., et al.: Domain adaptation from multiple sources via auxiliary classifiers. In: International Conference on Machine Learning. pp. 289–296 (2009)
5. Gu, T., Wu, Z., Tao, X., Pung, H., , Lu, J.: epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In: Conference on Pervasive Computing and Communication (2009)
6. Inomata, T., et al.: Activity recognition from interactions with objects using dynamic bayesian network. In: Context-Awareness for Self-Managing Systems. pp. 39–42 (2009)
7. van Kasteren, T., Englebienne, G., Krose, B.: Recognizing activities in multiple contexts using transfer learning. In: AAAI AI in Eldercare Symposium (2008)
8. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: VLDB. pp. 180–191 (2004)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
10. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37(1), 145–151 (1991)
11. Manning, C.D., Schtze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
12. Mozer, M.C., et al.: The neural network house: An overview. In: AAAI Spring Symposium on Intelligent Environments. pp. 110–114 (1998)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359 (2010)
14. Pérez, A., Larraòaga, P., Inza, I.: Bayesian classifiers based on kernel density estimation. Journal of Approximate Reasoning 50(2), 341–362 (2009)
15. Philipose, M., et al.: Inferring activities from interactions with objects. IEEE Pervasive Computing 3(4), 50–57 (2004)
16. Rashidi, P., Cook, D.J.: the resident in the loop: Adapting the smart home to the user. Systems, Man, and Cybernetics, Part A 39(5), 949–959 (2009)
17. Rashidi, P., Cook, D.J.: Transferring learned activities in smart environments. In: Conference on Intelligent Environments. vol. 2, pp. 185–192 (2009)
18. Rashidi, P., Cook, D.J.: Multi home transfer learning for resident activity discovery and recognition. In: KDD Knowledge Discovery from Sensor Data. pp. 56–63 (2010)
19. Rosenstein, M.T., et al.: To transfer or not to transfer. In: NIPS Inductive Transfer: 10 Years Later (2005)
20. Wyatt, D., Philipose, M., Choudhury, T.: Unsupervised activity recognition using automatically mined common sense. In: AAAI. pp. 21–27 (2005)
21. Zheng, V.W., Hu, D.H., Yang, Q.: Cross-domain activity recognition. In: Ubicomp. pp. 61–70 (2009)