Transfer Learning for Activity Recognition via Sensor Mapping

Derek Hao Hu and **Qiang Yang**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{derekhh, qyang}@cse.ust.hk

Abstract

Activity recognition aims to identify and predict human activities based on a series of sensor readings. In recent years, machine learning methods have become popular in solving activity recognition problems. A special difficulty for adopting machine learning methods is the workload to annotate a large number of sensor readings as training data. Labeling sensor readings for their corresponding activities is a time-consuming task. In practice, we often have a set of labeled training instances ready for an activity recognition task. If we can transfer such knowledge to a new activity recognition scenario that is different from, but related to, the source domain, it will ease our effort to perform manual labeling of training data for the new scenario. In this paper, we propose a transfer learning framework based on automatically learning a correspondence between different sets of sensors to solve this transfer-learning in activity recognition problem. We validate our framework on two different datasets and compare it against previous approaches of activity recognition, and demonstrate its effectiveness.

1 Introduction

Activity recognition is the problem of inferring and predicting human intentions and activities based on a series of sensor readings. Due to its multidisciplinary nature, it has accumulated lots of research interest from many areas including artificial intelligence, ubiquitous computing and sensor networks and also ranging to many areas including travel recommendation [Zheng and Xie, 2011], routine discovery [Farrahi and Gatica-Perez, 2011], conversation recognition [Hsu *et al.*, 2011], etc. Since the relationship between different sensor readings and activities is complex, machine learning algorithms have become popular in solving activity recognition tasks.

A common problem with supervised machine learning is the potentially expensive manual effort needed to label the training data. This problem is very pronounced in the field of activity recognition. One assumption required by most supervised learning methods is that the training and test data should be in the same feature space and have the same underlying distribution and the same label space. However, when the distributions and features are different between training and future data, the model performance often drops.

In the context of activity recognition, the above assumption manifests itself as: 1) The same feature space requirement means that training and testing data should use the same set of sensors; 2) The same underlying distribution requirement means that the preferences or the habit of the subjects should be similar in both training and testing data and 3) The same label space requirement means the activity set recognized in the training and testing data are the same.

To relax the assumption of same feature and label space as well as underlying distributions, many transfer learning algorithms have been developed to reduce labeling effort while still maintaining a reasonable accuracy . In transfer learning, useful knowledge from the source domain are being transferred to the target domain where labeled data is usually insufficient to build a reliable classifier on its own [Pan and Yang, 2010; Taylor and Stone, 2009]. Recently, researchers have tried to bring the idea of transfer learning into activity recognition, but but most of their approaches have certain associated limitations, which we will discuss in Section 2.

In this paper, we propose a transfer learning framework under which one can transfer the knowledge between different activity recognition tasks, relaxing the assumption of same feature space, same label space as well as same underlying distribution by automatically learning a mapping between different sensors. To build a mapping between the two domains, we use Web knowledge as a bridge to help link the different label spaces.

The rest of this paper is organized as follows. We will first introduce some related works on activity recognition and transfer learning. In Section 3, we will explain our proposed algorithm in detail. Next, we will evaluate our algorithm on a number of real-world activity recognition datasets to better demonstrate its effectiveness. Finally, we conclude this paper and point out several directions for our future research.

2 Related Work

Activity recognition aims to infer user behaviors from observations such as low-level sensor readings. However, most of the proposed activity recognition algorithms focused on sensor readings from only one domain, and usually require lots of annotated data to train the activity recognition model.

Transfer learning is motivated by the fact that humans can intelligently apply knowledge learned previously to solve new problems faster. Transfer learning has already been demonstrated successful in many scenarios [Pan and Yang, 2010] . More specifically, there have been several works that tries to link transfer learning with activity recognition [van Kasteren *et al.*, 2010; Zheng *et al.*, 2009]. In [Zheng *et al.*, 2009], the authors relax the same label space assumption by using Web knowledge as a bridge to transfer knowledge between different label spaces. The limitation in this paper is that the requirement of same feature space still applies, *i.e.*, the sensors in the source domain and the target domain should be the same. Such a limitation forbids many possible transferring scenarios in activity recognition. In [van Kasteren *et al.*, 2010], the authors studied activity recognition across different sensors. However, their algorithm is based on the usage of a *meta-feature space*, which are features that describe the properties of the actual features, *e.g.*, both sensors installed on microwaves and stoves have a meta feature as "kitchen heating". Each sensor is described by one or more meta features. The limitations of the approach described in [van Kasteren *et al.*, 2010] is that the meta-feature space needs to be manually constructed. Besides, different room layouts or different kinds of sensors would lead to huge difference in the meta features of the two rooms and hence the applicability of the algorithm is limited. Rashidi and Cook [Rashidi and Cook, 2011] studied transferring activity recognition knowledge from multiple source domains. However, their work does not allow the feature space of different domains to be different.

In our activity recognition problem setting, we need to transfer knowledge between different feature spaces, underlying distribution as well as different label spaces. In the transfer learning literature, transferring between different feature spaces has been studied extensively, *e.g.*, [Dai *et al.*, 2008; Yang *et al.*, 2009]. However, few research works have dealt with the knowledge transfer problem that involve changes in all of feature, distribution and label spaces.

3 Proposed Approach

We first define our *transfer learning for activity recognition* problem setting. We study two domains that have different sets of sensors and different activity labels. Specifically, we have a source domain where the labeled sensor readings are in the form of $\{(\mathbf{x_s}, \mathbf{y_s})\}$, and a target domain where we assume that we have only the unlabeled data the form of $\{(\mathbf{x_t}\}$. The source domain label space is defined as $\mathcal{L}_s$ and the target domain label space is defined as $\mathcal{L}_t$. We make the assumption that $\mathcal{L}_s$ and $\mathcal{L}_t$ are different, but are related through a probability function $p(y_s, y_t)$ where y_s and y_t are source and target-domain activity labels, respectively. This probability function between the label spaces can be learned by labeling some of the target domain instances, or through the Web (as we do in this paper).

Our final goal is to estimate $p(\mathbf{y_t}|\mathbf{x_t})$. We know that:

$$p(\mathbf{y_t}|\mathbf{x_t}) = \sum_{\mathbf{c}^{(i)} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}) \cdot p(\mathbf{y_t}|\mathbf{c})$$

Since the activity-label spaces $\mathcal{L}_s, \mathcal{L}_t$ may be large, for simplicity, in this paper, we approximate the value of $p(\mathbf{y_t}|\mathbf{x_t})$ by the *mode* (the most frequent label) of $p(\mathbf{c}|\mathbf{x_t})$, where $\mathbf{c}$ is an activity label, and denote the mode as $\hat{\mathbf{c}}$. $\hat{\mathbf{c}}$ is labeled using labels from the source domain $\mathcal{L}_s$. In other words,

$$p(\mathbf{y_t}|\mathbf{x_t}) \approx p(\hat{\mathbf{c}}|\mathbf{x_t}) \cdot p(\mathbf{y_t}|\hat{\mathbf{c}}) \quad (\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}))$$

In this paper, since we assume the two label spaces to be different but related, the joint distribution $p(\mathbf{y_s}, \mathbf{y_t})$ should have high mutual information in general. Therefore, $p(\mathbf{y_t}|\hat{\mathbf{c}})$ should also be high.

From the above equation, our transfer learning framework takes two steps. In the first step, we will estimate $p(\hat{\mathbf{c}}|\mathbf{x_t})$ where $\hat{\mathbf{c}}$ is labeled using the source domain label space $\mathcal{L}_s$. Briefly speaking, we aim to use the source domain label space to explain the target domain sequences $\mathbf{x_t}$ first. Since the two domains have different feature spaces, in our first step we need to transfer across different feature spaces.

Next, we estimate $p(\mathbf{y_t}|\hat{\mathbf{c}})$ where $\mathbf{y_t}$ is defined on the target domain label space $\mathcal{L}_t$ and $\hat{\mathbf{c}}$ is defined on the source domain label space $\mathcal{L}_s$; *i.e.*, in our second step, we need to transfer across different label spaces.

3.1 Transfer Across Feature Spaces

Based on the above discussions, in this section we first need to transfer knowledge between different feature spaces and estimate $p(\hat{\mathbf{c}}|\mathbf{x_t})$. For each sensor reading $\mathbf{x_s}$ in the source domain $\mathcal{S}$, $\mathbf{x_s}$ is represented by features $f_{\mathcal{S}}$. Similarly, for each sensor reading $\mathbf{x_t}$ in the target domain $\mathcal{T}$, denote the features composing $\mathbf{x_t}$ as $f_{\mathcal{T}}$. For example, $f_{\mathcal{S}}$ can be an on-body 3D accelerometer attached to the wrist and $f_{\mathcal{T}}$ can be a the Wifi signals from a mobile phone. In this section, we build a bridge between $f_{\mathcal{S}}$ and $f_{\mathcal{T}}$.

We use a framework similar to translated learning [Dai *et al.*, 2008]. When transferring the knowledge across different feature spaces, an important step is to find a *translator* $\phi(f_t, f_s) \propto p(f_t|f_s)$ (Here f_s and f_t are features of the data in $\mathcal{S}$ and $\mathcal{T}$, respectively.) between the source and target domains. Since f_t and f_s are conditionally independent given x_s, we have:

$$\begin{aligned} p(f_t, f_s) &= \int_{\mathcal{X}_s} p(f_t|x_s) p(f_s|x_s) p(x_s) dx_s \\ &= \int_{\mathcal{X}_s} p(f_t, x_s) p(f_s|x_s) dx_s \end{aligned}$$

In order to measure the joint distribution $p(f_t, f_s)$, we need to measure $p(f_t, x_s)$, or more precisely, the joint distribution between each feature in $\mathcal{T}$ with the source domain sensor readings $\mathbf{x_s}$. In order to measure this joint distribution, depending on whether we compute based on the difference on distributions or difference on signal data, we can use two basic tools to approximate $p(f_t, x_s)$: Jeffrey's J-divergence [Jeffreys, 1946] (the symmetric version of the KL-divergence) and Dynamic Time Warping [Keogh and Pazzani, 2000].

We can extract two kinds of information from sensor readings. The first is that, given a sequence of sensor reading, we can try to estimate the generative distribution from which such a sensor reading is generated. Since we only care about the relative distance between two distributions of sensor readings instead of describing these distributions in high accuracy, we simply plot the frequency of each sensor value (discretize the sensor value if it is continuous), and then smooth the discretized probability distribution. Since we have quite different feature spaces, we first normalize all our sensor readings into the range of [0,1].

In particular, suppose that we have a training set in the source domain $\{x_i, y_i\}$, where x_i are sensor readings and y_i are target labels. For each activity y_i, we can select all sequences of sensor readings x that have y_i as its label. Next, we would count the occurrences of sensor values x_{ij}, and then estimate the probability distribution for each of the sensor in the sensor reading sequence x_i. An intuitive explanation of the above-mentioned method is that we try to link each generative distribution of different sensors to a target activity. We could imagine that we are trying to compose a dictionary where words in this dictionary are in fact distributions of sensor readings, and we attempt to tell the readers that "if you encounter such a distribution in your sensor readings, then it is possible that the sensor readings correspond to such an activity".

Following a similar approach, we can also estimate the probability distribution for each sensor reading sequence in the target domain. Now that for each sensor reading sequence, we have an estimated distribution Q and we wish to find a close distribution P in the source domain. Since KL divergence is asymmetric, *i.e.* $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$. Therefore, instead of calculating $D_{KL}(P \parallel Q)$, we use $D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)$, which is undoubtedly a symmetric measurement, to measure the distance between two distributions generating sensor readings.

Two issues need addressed for the selection of candidate labels based on relative entropy measurements alone. The first is that, although $D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)$ equals to zero if and only if the two distributions P and Q are identical, the fact that sensors have a very large value does not necessarily mean the two distributions are highly uncorrelated. Consider two accelerometers where the directions of accelerations are different. In this case, whenever the first accelerometer senses a high value, the second accelerometer will sense a low value. Therefore, we need to consider distribution pairs at both high divergence and low divergence values.

The second issue we consider is the different sampling rates of different sensors when plotting their signal values versus time. Different kinds of sensors have very different sampling rates and the accuracy of distributions estimated can vary a lot. When calculating the correlation between different sensors, another important step is to use a distance metric that can take different sampling rates into account. Now given two series of sensor readings of only one dimension: Q and C of length n and m, we wish to align two sequences use dynamic time warping (DTW) [Keogh and Pazzani, 2000]. The idea of DTW is simple. We could construct an n-by-m matrix where the element at (i, j) contains the distance $d(q_i, c_j)$ between the two points q_i and c_j, which is measured as the absolute value of difference of q_i and c_j: $|q_i - c_j|$. Since each element (i, j) corresponds to the alignment between q_i and c_j. Our objective is to find a warping path W which is a contiguous set of matrix elements that defines mapping between Q and C. Thus, the element at position K of the warping path W is defined as $w_k = (i, j)_k$. This warping path can be found using dynamic programming under a quadratic time complexity.

Algorithm 1 Projecting the labels in the source domain to the unlabeled sensor readings in the target domain

```
Input: Source domain activities $\S_s$, source domain data $\mathcal{D}_s = \{(\mathbf{x}_s, \mathbf{y}_s)\} = \{(x_i, y_i) | y_i \in \mathcal{L}_s\}$, target domain data $\mathcal{D}_t = \{(\mathbf{x}_t)\}$
Output: Pseudo-labeled target domain data: $\mathcal{D}'_t = \{(\mathbf{x}_s, \mathbf{y}'_s\}$
begin
    Normalize each sensor reading sequence both in $\mathcal{S}$ and $\mathcal{T}$.
    For each pair of sensor reading and activity in $(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}$, estimate its probability distribution $p(f_s|y_s)$.
    For each unlabeled sequence in the target domain $\mathbf{x}_t$, estimate the distribution of its feature values: $P(f_t)$.
    Calculate the relative entropy between distributions in $\mathcal{T}$ and all the distributions in $\mathcal{S}$. Take the top-$K$ similar and the bottom-$K$ similar distributions out and record their labels as candidates.
    Calculate the DTW score between this sensor reading sequence $\mathbf{x}_t$ and all the labeled sensor reading sequences $(\mathbf{x}_s, \mathbf{y}_s)$ in the source domain. Take the top-$K$ highest and the bottom-$K$ lowest similar sensor readings out and record their labels as candidates.
    Label this unlabeled sequence $\mathbf{x}_t$ with the label that appeared maximum times in the candidate label set.
end
```

Algorithm 1 shows the step for projecting the labels in the source domain to the unlabeled sensor readings in the target domain. Notice that in this algorithm, we had introduced a parameter K, which is used to control the number of candidate label sequences in the source domain. In our experiments, we would test how variations of this parameter K would affect the overall algorithm performance.

3.2 Transfer Across Label Spaces

In our previous subsection, we had already estimated the value for $\arg\max_{\mathbf{c}} p(\hat{\mathbf{c}}|\mathbf{x_t})$. In this subsection, we aim to estimate $p(\mathbf{y_t}|\hat{\mathbf{c}})$, we have:

$$p(\mathbf{y_t}|\mathbf{c}) = p(\mathbf{y_t}, \mathbf{c})/p(\mathbf{c})$$

If we assume that there is no distinction between the prior distribution $p(\mathbf{c})$, then we have $p(\mathbf{y_t}|\mathbf{c}) \propto p(\mathbf{y_t}, \mathbf{c})$.

Based on the Markov assumption, we have:

$$\begin{aligned} p(\mathbf{y_t}, \mathbf{c}) &= p(y_t^0) \prod_i p(y_t^i | y_t^{i-1}) \prod_i p(c^i | y_t^i) \\ &\propto \prod_i p(y_t^i | y_t^{i-1}) \prod_i p(c^i | y_t^i) \\ \log p(\mathbf{y_t}, \mathbf{c}) &\propto \sum_i \log p(y_t^i | y_t^{i-1}) + \sum_i \log p(c^i | y_t^i) \end{aligned}$$

From the above formulation, we can see that such a problem can be reduced to estimating $p(l_s|l_t)$, where $l_s \in \mathcal{L}_s, l_t \in \mathcal{L}_t$ and $p(l_t^1|l_t^2)$, where $l_t^1, l_t^2 \in \mathcal{L}_t$. Since the number of labeled training data in the target domain is not sufficient, we need extra knowledge sources to estimate such probabilities.

For example, in [Shen *et al.*, 2006], the authors used Web pages from Open Directory Project (ODP) as a bridge to estimate the probabilities. In [Zheng *et al.*, 2009], the authors tried to calculate the cosine similarity of two word vectors, which are composed by the words of the Web search results when two activity names are used as queries and issued as input. In practice, such algorithms based on words from Web pages could be extremely slow. Instead of measuring the conditional probabilities directly, we choose to optimize a similar measurement that intrinsically can be optimized similarly as $p(\mathbf{y_t}, \mathbf{c}$, stated below.

We define $R(i, j)$ as the expected loss of assigning $j \in \mathcal{L}_t$ to y_t^i. $Q(l_1, l_2)$ as the "information distance" between l_1 and l_2, which are activity labels from the source and target domains, respectively. Then $R(i, j)$ is defined recursively as:

$$R(i,j) = \min_{k \in \mathcal{L}_t} \{R(i-1,k) + Q((\hat{\mathbf{c}})^i, j) + Q(k,j)\}$$

We briefly explain the nature of this recurrence relation. In order to minimize the loss up to time slice i, we need to first consider the minimum loss up to time slice $i-1$. We need to enumerate all possible $R(i-1, k)$, where $k \in \mathcal{L}_t$ is the label we assigned to time slice $i-1$. Next, we need to minimize the distance between the original "pseudo-label" $\hat{\mathbf{c}}^i$ and this new label $j \in \mathcal{L}_t$. Furthermore, $Q(k, j)$ is also considered in the loss function to minimize the distance between successive slices $\mathbf{y_t}^i$ and $\mathbf{y_t}^{i-1}$. It can be seen that the above recurrence relation could be solved using dynamic programming. In this paper, we use the Google Similarity Distance [Cilibrasi and Vitányi, 2007] as Q to approximate the information distance between two entities.

The definition of Google similarity difference is as follows:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

where $f(x)$ denotes the number of pages containing x, and $f(x, y)$ denotes the number of pages containing both x and y, as reported by Google. N is just a normalized factor that does not affect whether x is closer to y or z. Therefore, what we need to know is just a count of the search results. A detailed explanation of the *Google distance* is beyond the scope of this paper and we encourage readers to read [Cilibrasi and Vitányi, 2007] for technical details.

By using the normalized Google distance, our loss function becomes:

$$R(i,j) = \min_{k \in \mathcal{L}_t} \{R(i-1,k) + NGD((\hat{\mathbf{c}})^i, j) + NGD(k,j)\}$$

Algorithm 2 shows our transferring procedure. After these two steps, we now have the labels $y_t^i \in L_t$ for each unlabeled sensor reading in the target domain, and we can apply *any* machine learning algorithms used for activity recognition such as hidden Markov models (HMM) [Patterson *et al.*, 2005] or conditional random fields (CRF) [Vail *et al.*, 2007], to train activity recognition classifiers in the target domain.

4 Experimental Results

In this section, we investigate how our algorithm performs in several real-world activity recognition domains. Specifically, we test the recognition accuracy of our algorithm when transferring across different feature and label spaces.

Algorithm 2 Projecting target domain sequences with source domain labels to target domain sequences with target domain labels

```
Input: Pseudo-labeled target domain data $\mathcal{D}'_t = \{(\mathbf{x}_t, \hat{\mathbf{c}})\}$
Output: Labeled target domain data: $\mathcal{D}^*_t = \{(\mathbf{x}_t, \mathbf{y}_t)\}$
begin
    For each pseudo-labeled target domain instance $d'_t$, calculate its minimum loss value $R(i, j)$ based on the recurrence relation $R(i, j) = \min_{k \in \mathcal{L}_t}\{R(i-1, k) + NGD((\hat{\mathbf{c}})^i, j) + NGD(k, j)\}$, where $NGD$ is the Google similarity distance metric.
    Relabel this $d'_t$ using the labels in the target domain label space, thereby creating a new sequence $d^*_t$.
end
```

4.1 Datasets and Evaluation Criteria

In this paper, we use three real-world activity recognition datasets to validate our algorithm. Our first dataset (UvA in short) [1] is from [van Kasteren *et al.*, 2008] where a dataset is recorded in the house of a 26-year-old man, living alone in a three-room apartment where 14 state-change sensors are installed. The second dataset we use is the MIT PLIA1 dataset [2] [Intille *et al.*, 2006], which was recorded on March 4, 2005 from 9AM to 1PM in the MIT PlaceLab. The third dataset is from [Patterson *et al.*, 2005] (Intel in short), which aims to recognize 12 routine morning activities based on RFID sensors.

4.2 Baseline

To allow a better comparison of our algorithm performance against state-of-the-art research in activity recognition, we compared against an unsupervised activity recognition algorithm described in [Wyatt *et al.*, 2005]. Briefly, in [Wyatt *et al.*, 2005] describes an unsupervised activity recognition algorithm that can infer the activities being performed based on object names involved in the activities. Notice that, since algorithms described in [Zheng *et al.*, 2009] and [van Kasteren *et al.*, 2010] have different problem settings compared to our paper, (the former assumes different label space but same feature space and the latter assumes the meta-feature space is constructed manually), we cannot use their algorithms as baselines for comparison.

4.3 Different Features and Same Labels

In our first experiment, we aim to validate the effectiveness of our algorithm when transferring knowledge between different feature spaces. More precisely, in all of the three datasets we used, we divide the feature space into two. Half of the sensor readings are used as data in the source domain and the remaining half are used as data in the target domain. Since this split of source and target domains are done manually, the label space is still the same.

[1] http://staff.science.uva.nl/∼tlmkaste/research/software.php

[2] http://architecture.mit.edu/house_n/data/PlaceLab/PLIA1.htm

K	UvA Acc(Var)	Intel Acc(Var)
K = 5	55.8% (5.1%)	52.1% (4.7%)
K = 10	58.2% (4.3%)	53.4% (4.5%)
K = 15	67.3% (4.1%)	55.3% (3.8%)
K = 20	**68.2% (4.0%)**	**57.2% (4.2%)**
Unsupervised	47.3%(4.1%)	42.8% (3.8%)

Table 1: Algorithm Performance on UvA and Intel Dataset

Table 1 shows our algorithm performance on the UvA dataset and Intel dataset. We have repeated the splitting process for ten times and both the average accuracy and the variance are reported. We also report our algorithm performance by varying the parameter K. Recall that K is the parameter we use to control our candidate "label set size". More precisely, we select both the top-K similar sensor reading distributions and bottom-K dissimilar sensor reading distributions, as well as the top-K minimum DTW score sensor reading sequences, a total of $3K$ sensor readings in the source domain and their corresponding labels in the source domain, and put these labels in the candidate set.

Our result in Table 1 shows that our algorithm could consistently outperform the unsupervised activity recognition approach. We also observe that that with the increase of K, the accuracy also increases whereas the variance is also consistently decreasing. This is due to the fact more candidate labels are taken into account and therefore we could expect to consider more "probable" labels and our assignment of labels could be more precise. However, when K is larger than 20, performance starts to converge and also drops slightly. Therefore, we end by reporting our best result, which is achieved at $K = 20$.

89 activities are included in the MIT PLIA1 dataset and a taxonomy could be built to describe these activities [Intille *et al.*, 2006]. In MIT PLIA1 dataset, we analyze how the performance will be when we use the activities under the same category as both the sensor readings in the source domain and in the target domain. The same splitting process that was applied to the UvA and the Intel datasets is also applied to the sensor readings under each category to split the source domain and the target domain. The MIT PLIA1 dataset can be categorized into 9 subcategories, including cleaning indoor, yardwork, laundry, dishwashing, meal preparation, hygiene, grooming, personal and leisure. We report the accuracy and variance our algorithm had achieved in each subcategory in Table 2.

From Table 2, we can also see that our transfer learning activity recognition algorithm outperforms the unsupervised baseline in most cases and achieves comparable performance with the unsupervised baseline in other subcategories. For the choice of parameter K, we could see it exhibits a very different behavior as in Table 1. Generally speaking, the best performance is usually achieved when K is small. One possible explanation for this phenomenon is that in MIT PLIA1 dataset, since the dataset size is relatively large, the probability distribution estimated is relatively more accurate than the UvA or the Intel dataset, and therefore it is possible to achieve a much better performance with a smaller K. However, when K is larger, more noisy sensor readings are induced.

4.4 Different Features and Labels

In this experiment, we use the full MIT PLIA1 dataset as the source domain and then try to transfer to both the UvA dataset and the Intel dataset. We use such a way to validate our algorithm since we believe the direction of "transfer" is especially important since the size of UvA or Intel dataset will not contain enough knowledge from which we could transfer to the MIT dataset. Since the dimension of the feature space (number of sensors) and the dimension of the label space (number of activities) in MIT PLIA1 dataset are both significantly larger than those of UvA and Intel datasets, we choose to transfer from PLIA1 to UvA and Intel.

K	MIT → UvA Acc(Var)
K = 5	**59.8% (4.2%)**
K = 10	57.5% (4.1%)
K = 15	51.0% (4.8%)
K = 20	41.0% (4.1%)
Unsupervised	47.3%(4.1%)

Table 3: Algorithm performance of transferring knowledge from MIT PLIA1 to UvA dataset

K	MIT → Intel Acc(Var)
K = 5	60.5% (4.2%)
K = 10	**61.2% (3.8%)**
K = 15	53.2% (4.1%)
K = 20	42.0% (2.5%)
Unsupervised	42.8%(3.8%)

Table 4: Algorithm performance of transferring knowledge from MIT PLIA1 to Intel dataset

The results in Table 3 and 4 have validated that our approach of transferring knowledge across feature space and label space is effective.

5 Conclusion and Future Work

In this paper, we have proposed an approach to solve the activity recognition problem under the transfer learning setting. By comparing our method with many previous solutions which also attempt to solve the activity recognition problem under a transfer learning setting, we can see that our method does not have many of the limitations which have been encountered in previous papers. The fundamental assumption of our paper by transferring the knowledge across different feature spaces is that although the kinds of sensors we may encounter are highly different, the distributions may be similar and we can exploit such knowledge for building a bridge across domains. Furthermore, when handling the case of different label space across two domains, we can alleviate this problem to estimating the conditional distribution of two label spaces and then use Web knowledge as a tool to help estimate such a value. We have validated our approach in several real-world sensor-based activity recognition tasks and have

Category	K = 5 Acc (Var)	K = 10 Acc (Var)	K = 15 Acc (Var)	K = 20 Acc (Var)	Unsupervised Acc (Var)
Cleaning Indoor	**68.5%(2.4%)**	62.5%(2.9%)	61.8%(2.8%)	64.4%(3.0%)	50.7%(2.6%)
Yardwork	52.3%(2.9%)	**69.1%(3.9%)**	55.8%(2.8%)	52.7%(2.6%)	51.8%(2.1%)
Laundry	**69.3%(3.3%)**	60.6%(2.3%)	50.5%(2.9%)	63.6%(2.8%)	56.8%(3.2%)
Dishwashing	51.9%(2.6%)	**69.7%(3.6%)**	57.8%(3.5%)	66.4%(3.7%)	68.2%(2.9%)
Meal Preparation	53.3%(3.1%)	63.7%(3.0%)	64.4%(2.9%)	60.4%(2.1%)	**68.6%(2.9%)**
Hygiene	56.3%(2.5%)	**62.4%(3.7%)**	52.6%(2.9%)	57.9%(2.5%)	54.2%(3.6%)
Grooming	59.6%(3.9%)	65.1%(3.0%)	57.2%(2.7%)	62.0%(2.0%)	**69.9%(2.7%)**
Personal	**68.6%(3.2%)**	59.2%(3.4%)	57.8%(2.3%)	53.3%(3.4%)	68.2%(3.1%)
Leisure	59.3%(3.9%)	57.2%(3.2%)	65.3%(2.6%)	61.5%(2.3%)	**65.9%(3.2%)**

Table 2: Algorithm Performance on MIT PLIA1 Dataset

demonstrated the effectiveness of our algorithm compared to unsupervised activity recognition methods.

We plan to extend our work in the following directions. Firstly, we wish to study the detailed constraints under which our algorithm would work. Stating the correlation between source domain and target domain as "different but related" is difficult to judge in reality. Setting an accurate distance metric or constraint would be nicer for end users to judge whether the two domains can be used for transfer. Secondly, when we perform the transfer step, we have used $\hat{\mathbf{c}}$ as an approximation of all possible labeling sequences from the source domain label space. Different from other machine learning methods which use modes to approximate integrals or summations, the approximation ratio of such a method is not satisfactory. In our future study, we plan to choose a candidate pseudolabeling set, instead of a $\hat{\mathbf{c}}$ alone, to study the effect of transfer.

Acknowledgment

We thank the support of Hong Kong RGC/NSFC Project N_HKUST624/09 and Hong Kong RGC Project 621010.

References

[Cilibrasi and Vitányi, 2007] Rudi Cilibrasi and Paul M. B. Vitányi. The google similarity distance. *IEEE Trans. Knowl. Data Eng.*, 19(3):370–383, 2007.

[Dai *et al.*, 2008] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. Translated learning: Transfer learning across different feature spaces. In *NIPS*, pages 353–360, 2008.

[Farrahi and Gatica-Perez, 2011] Katayoun Farrahi and Daniel Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM TIST*, 2(1):3, 2011.

[Hsu *et al.*, 2011] Jane Yungjen Hsu, Chiachun Lian, and Wanrong Jih. Probabilistic models for concurrent chatting activity recognition. *ACM TIST*, 2(1):4, 2011.

[Intille *et al.*, 2006] Stephen S. Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Proceedings of the Fourth International Conference on Pervasive Computing (Pervasive 2006)*, pages 349–365, 2006.

[Jeffreys, 1946] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1946.

[Keogh and Pazzani, 2000] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD*, pages 285–289, 2000.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[Patterson *et al.*, 2005] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers (ISWC 2005)*, pages 44–51, 2005.

[Rashidi and Cook, 2011] Parisha Rashidi and Diane Cook. Activity knowledge transfer in smart environments. *Pervasive and Mobile Computing*, 2011.

[Shen *et al.*, 2006] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building bridges for web query classification. In *SIGIR*, pages 131–138, 2006.

[Taylor and Stone, 2009] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[Vail *et al.*, 2007] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 1–8, 2007.

[van Kasteren *et al.*, 2008] Tim van Kasteren, Athanasios K. Noulas, Gwenn Englebienne, and Ben J. A. Kröse. Accurate activity recognition in a home setting. In *UbiComp*, pages 1–9, 2008.

[van Kasteren *et al.*, 2010] Tim van Kasteren, Gwenn Englebienne, and Ben J. A. Kröse. Transferring knowledge of activity recognition across sensor networks. In *Pervasive*, pages 283–300, 2010.

[Wyatt *et al.*, 2005] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *AAAI*, pages 21–27, 2005.

[Yang *et al.*, 2009] Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the socialweb. In *ACL/AFNLP*, pages 1–9, 2009.

[Zheng and Xie, 2011] Yu Zheng and Xing Xie. Learning travel recommendations from user-generated gps traces. *ACM TIST*, 2(1):2, 2011.

[Zheng *et al.*, 2009] Vincent Wenchen Zheng, Derek Hao Hu, and Qiang Yang. Cross-domain activity recognition. In *UbiComp*, pages 61–70, 2009.