

Activity Recognition Using Transfer Learning

Wen-Hui Chen,^{*} Po-Chuan Cho, and Yong-Lin Jiang

Graduate Institute of Automation Technology, National Taipei University of Technology
No. 1. Sec. 3, Zhongxiao E. Rd., Taipei, Taiwan

(Received April 1, 2016; accepted February 28, 2017)

Keywords: activity recognition, machine learning, transfer learning, smart homes

The technology for human activity recognition has become an active research topic in recent years as it has many potential applications, such as surveillance systems, healthcare systems, and human-computer interaction. In the research of activity recognition, supervised machine learning approaches have been widely used for activity recognition. However, the cost of collecting labeled sensor data in new environments is high. Furthermore, these methods do not work well in a cross-domain environment using conventional machine learning approaches. In this study, we proposed a transfer learning framework based on principal component analysis (PCA) transformation, Gale–Shapley similarity measurement, and Jensen–Shannon divergence (JSD) feature mapping. Transfer learning aims to apply new information learned from the source domain to the target domain. The experimental results showed that the proposed approach performs better than the approach merely learned in the source environment.

1. Introduction

With the advance of sensor technology and machine learning algorithms, activity recognition has become an active research topic in recent years. Activity recognition can be applied to assisted living, human–computer interactions, and healthcare, especially for the elderly. In the activity recognition community, most researchers tend to use machine learning methods to tackle activity recognition problems. To determine the model parameters, collecting sufficient labeled data is required in the training process, and significant effort is required when applying the model to different environments. Most past research assumed that the distribution of the sensor data from new environments was the same as that used in the model training process. However, this assumption is not always valid and is difficult to satisfy in practical situations.

In this study, we proposed a transfer learning framework to overcome these problems. The proposed framework consists of three steps: (1) extract seminal information from the source and target environments with principal component analysis (PCA) transformation, (2) measure the feature similarity with the Jensen–Shannon divergence (JSD) algorithm, and (3) map features to a common space based on the Gale–Shapley algorithm. The proposed transfer framework can be used to increase the recognition performance when the model is applied to a new environment where the training sample is insufficient. In addition, it can also reduce the effort of obtaining labeling data.

^{*}Corresponding author: e-mail: whchen@ntut.edu.tw
<http://dx.doi.org/10.18494/SAM.2017.1546>

The rest of the paper is organized as follows. In Sect. 2, we provide a brief overview of activity recognition and the concept of transfer learning. The proposed transfer learning framework is described in Sect. 3. In Sect. 4, the results of two experiments using two publicly available datasets are discussed. Finally, conclusions are presented in Sect. 5.

2. Related Work

2.1 Activity recognition algorithms

Many approaches have been proposed in the study of the activity recognition problem. A survey of human activity recognition can be found in Ref. 1. Among the approaches, hidden Markov models (HMMs),^(2–4) naive Bayes classifiers (NBCs),⁽⁵⁾ decision trees,⁽⁶⁾ and support vector machines (SVMs),^(7,8) are widely used in activity recognition.

When an HMM is used in activity recognition, activities are the hidden states and can be recognized through a trained model. Although it has good recognition ability for continuous or regular behavior, the HMM approach requires two independent assumptions for tractable inference. NBCs are a classical classification method based on Bayesian theory. They have worked well in some areas, but their performance may not be as good as other classification algorithms due to the strong independent assumption regarding sensor features in activity recognition.

Decision trees are also a commonly used algorithm for classification problems. A decision tree algorithm has a flowchart-like structure, and its inference procedure is easy to understand. However, it is difficult for the algorithm to recognize complicated activities.

SVM is a supervised learning algorithm. The main spirit of SVM is to find the best hyperplane that can best separate different classes of data in high-dimensional feature space. With kernel function mapping, SVM can provide a robust solution in activity recognition.

2.2 Transfer learning

Most activity recognition approaches mentioned in Sect. 2.1 can perform well under the assumption that sensor data from the source and target domains are in the same distribution. However, in practice, different environments or sensor types can degrade the recognition performance. The transfer learning approach can overcome this issue by mapping features from source and target environments to a common space.

To date, a number of studies have been reported about activity recognition in a smart home setting using the concept of transfer learning. A survey for reviewing activity recognition based on transfer learning has been published.⁽⁹⁾

Kasteren *et al.*⁽¹⁰⁾ proposed three different feature mapping functions called function groups to project sensor features to a common space. After projecting, a semisupervised hidden Markov model and improved expectation-maximization (EM) algorithm were adopted for activity recognition. In addition, Kasteren *et al.* also adopted the mapping of sensor data from different environments into an individual feature space called meta-feature space as described in Ref. 11. Handling mapping relationships can reduce the dimension of features and the cost of mapping computations. Each individual model is combined according to previously assigned weights. In Ref. 12, the authors applied their background knowledge of sensors such as the deployed locations, types of sensors, mounted objects, and triggering events to assign a weight for each sensor. According to the designated weight, they proposed an approach that can carry out sensor

matching between two different environments to achieve the transfer of knowledge without any target domain data. Instead of using the background knowledge of sensors, Rashidi and Cook⁽¹³⁾ proposed an iterative parameter updating algorithm with a semi-EM algorithm called home-to-home transfer learning (HHTL). By iteratively updating the matrix that describes links between sensors and activities, their model can perform activity transfer from a source space to a target space.

3. The Proposed Approach

A major problem encountered in current activity recognition research is the requirement of collecting sufficient labeled data in the target environment to train classification models. In practice, system developers may be able to obtain sufficient labeled data from the source environment but can rarely acquire sufficient data from the target environment where the system will be deployed. In this situation, training samples are insufficient to train a reliable model that can be applied to the target environment. In this study, we proposed a transfer learning framework that can map features in the source space and target space into a common space to overcome this problem.

The diagram of the proposed framework is shown in Fig. 1. First, features in the source and target environments are transformed to a space with a higher divergence that contains more independent information. Next, we estimate the similarity between each feature as the reference for feature mapping. To insure a one-on-one mapping relationship, the Gale–Shapley algorithm is adopted. Finally, a common feature space is built, and all features are mapped onto the common space for training and testing. The details of each step are described in the following sections.

3.1 Feature linear coordinate transformation

The procedure of feature linear coordinate transformation is the first step for the proposed transfer learning framework. The transformation procedure should follow two rules: (1) each

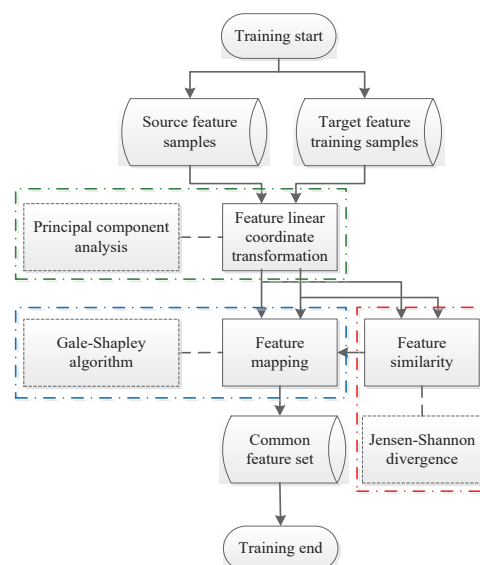


Fig. 1. (Color online) Proposed transfer learning framework.

feature will provide highly divergent information, and (2) the lost information will be minimized. A dimensionality reduction based on PCA is used to complete the transformation.

PCA can be used to achieve linear dimension reduction. In this study, we take the activity labels as an input feature vector and compute the expected value of covariance matrices as shown in Eq. (1). The top ten percent of the derived eigenvalues are selected, and their corresponding eigenvectors are adopted for dimension reduction.

$$E[Cov(AX|Z)] = E[ACov(X|Z)A^T] = A\Sigma_x A^T \quad (1)$$

3.2 Feature similarity

When measuring feature similarity, the similarity between the source and the target features is easy to compute if they are exactly the same. However, they are usually different in practical situations.

There are various methods for similarity measurement, such as mutual information, correlation coefficient, and Euclidean distance. In this study, we use JSD to estimate the similarity between two domain features. JSD is a probability-based algorithm suitable for measuring two probability distributions. It is an improved algorithm based on the Kullback–Leibler divergence (KLD) and can be used to solve a non-symmetric and unbounded problem of KLD. The basic KLD is formulated in Eq. (2), where P and Q are two probability functions. The formulation of JSD is shown in Eq. (3), where $R = (P + Q)/2$. As can be observed, JSD is symmetric, and the boundary can be converged in $[0, 1]$ when the logarithm base is set to 2.

$$KLD(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \quad (2)$$

$$JSD(P\|Q) = \frac{1}{2} [KLD(P\|R) + KLD(Q\|R)] \quad (3)$$

To take the activity labels into account, we separately calculate the JSD with different labels. The results are summed to estimate the overall JSD of the features. If we assume that F and G are two feature spaces, we can calculate the expected value of the JSD with Eq. (4), where P_i and Q_m are the distributions of $f_i \in F$ and $g_m \in G$, respectively, Z is the activity label, and f_i is the probability distribution function of the label Z .

$$E\{JSD[P_i(f_i|Z)\|Q_m(g_m|Z)]\} = \sum_k f_i(z_k) JSD[P_i(f_i|Z = z_k)\|Q_m(g_m|Z = z_k)] \quad (4)$$

3.3 Feature mapping

The procedure of feature mapping aims to pair a one-on-one link between the source and the target features by finding the similarity with the highest value. This problem can be formatted as a graph matching problem. In graph theory, two vertexes can be connected with an edge, but no two edges can share one vertex. We consider the features of two different environments to be vertexes, and the edge represents the link between the features.

In the Gale–Shapley algorithm, features can be considered as a marriage relationship. The Gale–Shapley algorithm can find a stable link from the relationship. We treat the features in the

source domain as men and the features in the target domain as women. There are three cases that can occur in the pairing process: (1) features with no pair, (2) features with one-on-one pairs, and (3) features with multiple pairs. The action of the Gale–Shapley algorithm will hold features to wait in case 1, confirm the link in case 2, and pick one of the features by measurement values in case 3. As shown in Fig. 2, sensors can obtain exactly one-on-one pairs through iterative computation. Finally, we can complete feature mapping in the source and target domains and project them to a common space.

4. Experimental Results

4.1 Datasets

In this study, two widely used, publicly available datasets were adopted in our experiments: MAS622J from Massachusetts Institute of Technology (MIT),⁽¹⁴⁾ and the dataset adopted from Kasteren *et al.*⁽¹¹⁾

The summary of the first dataset is listed in Table 1. It contains two different environments denoted as MAS-S1 and MAS-S2, and activity records for sixteen days. We choose five common activities defined in MAS-S1 and MAS-S2 to verify our framework. The five selected activities are frequently performed by senior people, including using the toilet, preparing breakfast, preparing lunch, preparing dinner, and washing dishes.

The second dataset contains three houses denoted as House-A, House-B, and House-C. Each house has a different layout, and the activities performed by the subjects in these three houses are different. The sensors, such as switches, pressure mats, mercury contacts, and passive infrared (PIR) that are installed in these houses vary, as listed in Table 2. We also select eight common activities in the second dataset.

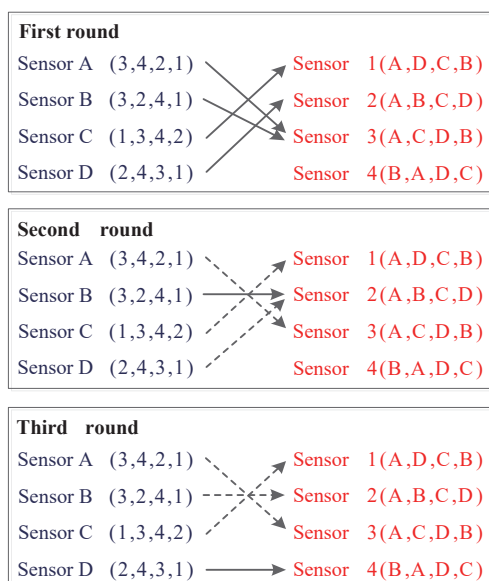


Fig. 2. (Color online) Feature mapping via the Gale–Shapley algorithm.

Table 1
The MIT Dataset.

Datasets	MAS-S1	MAS-S2
Number of sensors	76	70
Number of activities	13	9

Table 2
The Kasteren dataset.

Datasets	House-A	House-B	House-C
Number of residents	1	1	1
Number of rooms	2	3	6
Record time (d)	23	14	21
Layout	Apartment	Apartment	House

4.2 Experimental setup

To test the recognition accuracy, we applied the improved PCA to perform feature transformation. The feature similarity between two different domains can be derived by computing the expected value of JSD. Then, we applied the Gale–Shapley algorithm to find the best matching pair to achieve knowledge transfer. Finally, we built an SVM model with the radial basis kernel function to classify activities.

Before data processing, raw data must be encoded into a feature vector. We separate sensor data by 30 seconds without interval overlapping. If a sensor is triggered at an interval, its label will be set to 1 or 0. Furthermore, the trigger time, duration, and sensor location are also recorded.

To evaluate the recognition accuracy, we use the F-score or F-measure for performance measurement. The results contain four scores, including true positive (TP), false positive (FP), false negative (FN), and true negative (TN) for each class in a confusion matrix. The precision score, recall score, and F-measure are calculated using these four parameters as shown in Eqs. (5)–(7).

$$Precision = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (5)$$

$$Recall = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i} \quad (6)$$

$$F\text{-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

4.3 Experimental results for MIT dataset

In the first experiment, we tested the performance between the transform learning model and the model learned in the source environment. In the MIT dataset, we used all data from S1 and the first ten days of data from S2 for training. The remaining six days of data from S2 were regarded as the target domain for testing. Figure 3 shows a comparison between the use with and without transfer learning in the case of MIT datasets.

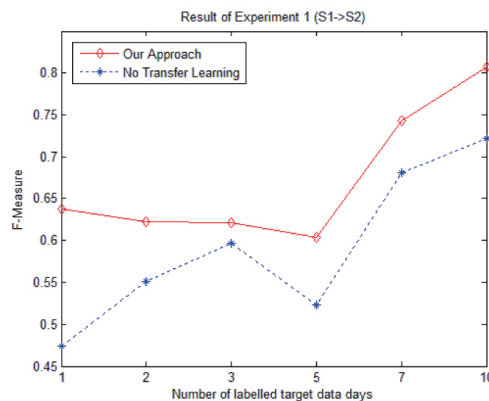


Fig. 3. (Color online) A comparison of recognition accuracy with and without transfer learning in the case of MIT datasets.

the proposed transfer learning approach. The x -axis represents the number of days that data was used in dataset S2 during the training process, while the y -axis represents the accuracy in terms of the F-measure.

From Fig. 3, we can observe when the training data is insufficient, and when the use of transfer learning is more accurate than the model without transfer learning. With the increase in the number of training samples, the recognition accuracy increases significantly from day 7 to day 10 for both the transfer and nontransfer models. Compared with the approach proposed in Ref. 12, our approach obtains an F-measure of 0.77, more accurate than 0.66 in the mapping from the dataset S1 to the dataset S2.

4.4 Experimental results for Kasteren dataset

In the second experiment, we compare our approach with other transfer learning approaches proposed by Kasteren *et al.* in Ref. 11. Kasteren *et al.* proposed two transfer learning frameworks: the single model and the separate model. In the single model framework, the authors used only one model for all training data. In the separate model, the authors obtained the model parameters for the source and the target environments and combined them with prior weights.

In our experimental setting, the sensor data stream was divided by a time frame of sixty seconds as an interval, and eight common activities were used in three house settings. We selected one house as the source environment and the other as the target environment. The data collected in days one to ten from the target domain were sequentially added to the training samples, and the rest of the data were set as the testing samples. The results from each house are shown in Figs. 4(a) and 4(b).

In the case of House-A, the separate model proposed by the authors in Ref. 11 works well. The separate model can reduce the error caused by the difference in activity patterns and house layouts by taking the differences into consideration in each house to adjust the model parameters to improve the recognition accuracy. The performance of our approach also shows good results as shown in Fig. 4(a).

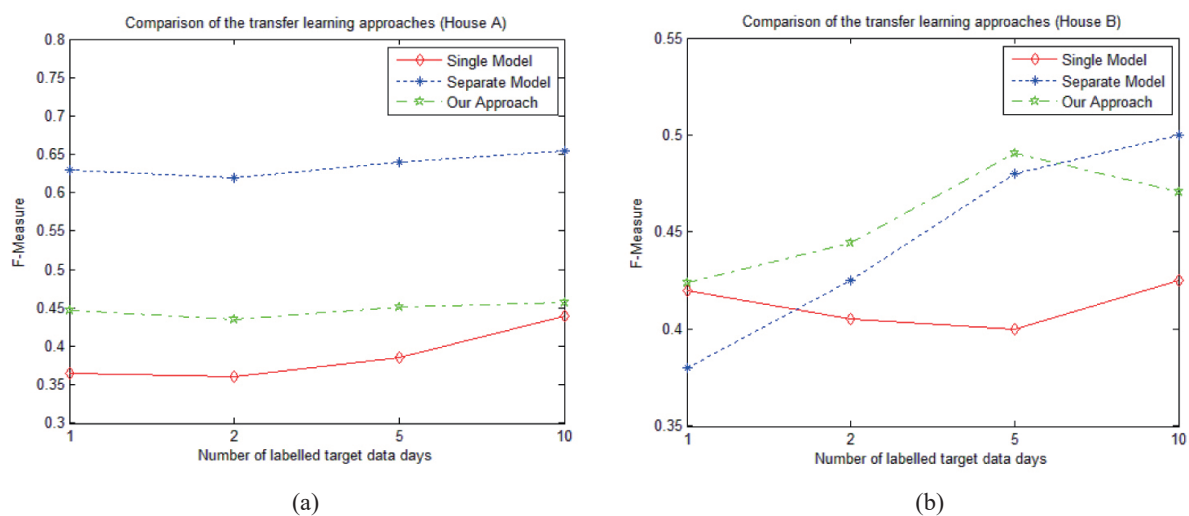


Fig. 4. (Color online) A comparison of recognition accuracy in the case of Kasteren datasets: (a) House-A testing results and (b) House-B testing results.

In the case of House-B, our framework obtains the best performance compared with the models proposed by Kasteren *et al.* with training samples collected in days one to five as shown in Fig. 4(b). The layout of House-B is a subdivision containing separable information. Although our approach has a better performance with insufficient data than the separate model, the performance of the separate model progressively improves with the increase in the number of training samples. The recognition accuracy of the single model is the poorest in both cases.

5. Conclusions

Activity recognition plays an important role in many applications such as smart homes, human–computer interactions, and elderly care. In this study, we have demonstrated the application of transfer learning to activity recognition in a smart home setting. With the use of transfer learning, we can take the benefits of reducing data labeling and improve recognition performance compared with that using only the training data from the source environment.

At first, we used the PCA algorithm to perform data transformation. Next, we estimated the feature similarity between the source and the target domains with the JSD algorithm. Finally, we used the Gale–Shapley algorithm to map sensor features to a common feature space to complete the knowledge transformation process. The preliminary experimental results showed that the proposed transfer learning framework can construct a better recognition model in a new environment than the traditional supervised learning model.

Acknowledgments

This work is supported by Taiwan Ministry of Science and Technology (MOST) under Grant No. MOST 104-2221-E-027-028.

References

- 1 L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu: *IEEE Trans. Syst. Man Cybern. Part C.* **23** (2012) 90.
- 2 A. Mannini and A. M. Sabatini: *Computational Intelligence and Neuroscience* (Hindawi, Italy, 2012) pp. 1–10.
- 3 G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Handel: *IEEE Trans. Instr. Meas.* **62** (2013) 1073.
- 4 C. Zhang and L. Zhang: *Int. J. Smart Home* **7** (2013) 237.
- 5 T. V. Kasteren and B. Krose: *Proc. Third IET Int. Conf. Intelligent Environments* (2007) pp. 209–212.
- 6 J. Pärkkä, L. Cluitmans, and M. Ermes: *IEEE Trans. Inf. Technol. Biomed.* **14** (2010) 1211.
- 7 A. Fleury, M. Vacher, and N. Noury: *IEEE Trans. Inf. Technol. Biomed.* **14** (2010) 274.
- 8 O. Banos, M. Damas, H. Pomares, F. Rojas, B. Delgado-Marquez, and O. Valenzuela: *Soft Comput.* **17** (2013) 333.
- 9 D. Cook, K. D. Feuz, and N. C. Krishnan: *Knowledge Inf. Syst.* **36** (2013) 45.
- 10 T. van Kasteren, G. Englebienne, and B. Kröse: *Proc. AAAI Fall Symp. AI in Eldercare: New Solutions to Old Problems.* (AAAI Press, Chicago, 2008) pp. 1–8.
- 11 T. van Kasteren, G. Englebienne, and B. Kröse: *Proc. 8th Int. Conf. Pervasive Computing* (Heidelberg, Germany, 2010) pp. 283–300.
- 12 Y. T. Chiang and J. Y. Hsu: *Proc. 9th Int. Conf. Ubiquitous Intelligence & Computing and 9th Int. Conf. Autonomic & Trusted Computing* (Fukuoka, Japan, 2012) pp. 180–187.
- 13 P. Rashidi and D. J. Cook: *Proc. the AAAI Plan, Activity and Intent Recognition Workshop* (Menlo Park, USA, 2010) pp. 45–52.
- 14 E. M. Tapia: *Master of Science Thesis, Massachusetts Institute of Technology* (2003).