An Asynchronous Multi-View Learning Approach for Activity Recognition using Wearables

Yuchao Ma[1] and Hassan Ghasemzadeh[1]

***Abstract*— In this paper, we introduce an Asynchronous Multiview Learning (AML) approach to allow accurate transfer of activity classification models across asynchronous sensor views. Our study is motivated by the highly dynamic nature of health monitoring using wearable sensors. Such dynamics include changes in sensing platform (e.g., sensor upgrade) and platform settings (e.g., sampling frequency, on-body sensor location), which result in failure of the machine learning algorithms if they remain untrained in the new setting. Our approach allows machine learning algorithms to automatically reconfigure without any need for labeled training data in the new setting. Our evaluation using real data collected with wearable motion sensors demonstrates that the average classification accuracy using our automatically labeled training data is 85.2%. This accuracy is only 3.4% to 4.5% less than the experimental upper bound, where ground truth labeled training data are used to develop a new activity recognition classifier.**

I. Introduction

Wearables, leveraging with computational algorithms, have emerged as a revolutionary technology in many application domains in healthcare and wellness [1], [2]. The most commonly used computational algorithms for such applications are based on supervised machine learning techniques. Retraining of these algorithms requires collecting a sufficient amount of labeled training data, it's a time consuming, labor-intensive, and expensive process that has been identified as a major barrier to personalized medicine [3]. This problem becomes more challenging considering the highly dynamic deployment of wearable equipments and rapid upgrade of the sensor platforms.

One common scenario is, having a training dataset of human activities collected using an old platform, to perform activity recognition for unlabeled dataset collected in a new sensor platform utilizing the labeled training data from the old view. Directly applying previously-trained machine learning algorithms on the new unlabeled dataset will result in a dramatic decline of the classification accuracy.

In this paper, we introduce an activity recognition approach, Asynchronous Multi-view Learning (AML), for knowledge transfer between distinct sensor views where the old and new sensors do not sample the physical activity simultaneously. At the core of our AML algorithm is an inter-feature mapping (IFM) method to facilitate information learning by associating instances and attributes of the two datasets collected in two different views.

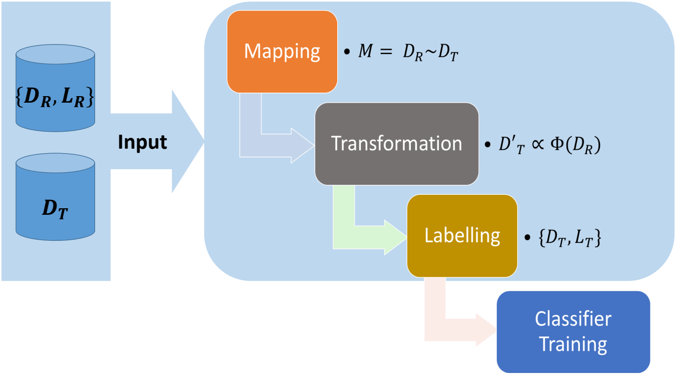


Fig. 1. Overview of the Proposed AML Framework

II. Transfer Learning Framework Overview

We refer to an old platform as 'source' and a new one as 'target'. An arbitrary application for sensor data collection and feature extraction works in the same way on both platforms, but we do not have access to its internal parameters. We assume that only a labeled dataset from 'source' is available, and that 'target' can sample the activity to generate an unlabeled dataset. Furthermore, we assume that the set of activities of interest to be identified is known *priori*.

Given a set of activity labels $L \in \{1, 2, \ldots, m\}$, a reference/source view R with a dataset D_R and corresponding labels L_R, and a target view T, the problem of *Asynchronous Multi-View Learning (AML)* is to construct a labeled training dataset in T and develop a classification algorithm for T such that the misclassification error is minimized.

To identify activities in the target view T by exploiting the labeled data from the asynchronous source view R, we first collect data in the target view and build an unlabeled dataset D_T. Fig. 1 illustrates the architectural overview of the proposed solution framework.

A. Mapping

Although the features computed in D and T are of the same type, the distribution of the features in the two feature spaces of D_R and D_T can be entirely different. Furthermore, the ordering of the features can be different because physical attributes of the two platforms can be different. Therefore, the first phase in our algorithm is to appropriately map each feature f_R^i in D_R to one feature f_T^j in D_T such that they refer to the same actual feature (e.g., they both refer to standard deviation of X-axis accelerometer) in feature set

[1] Y. Ma and H. Ghasemzadeh are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA (phone: 509-335-8260; e-mail: yuchao.ma@wsu.edu, hassan@eecs.wsu.edu).

f_S. In Section III, we present our inter-feature mapping algorithm in detail.

B. Transformation

Given a matching M from the feature space in D_R to that of D_T, we transform D_T to a new dataset D'_T with a similar format as D_R. There are two steps to obtain D'_T: (1) to adjust the feature ordering in D_T so that i_{th} feature in D'_T refers to the same one as the i_{th} feature in D_R; (2) to rescale each feature in D'_T such that the mean value of $f^i_{T'}$ equal to the mean value of the matched feature in D_R, as defined in the following equation.

$$f^i_{T'} = f^i_{T'} \times \frac{\mu(f^i_R)}{\mu(f^i_{T'})} \tag{1}$$

C. Labeling

To label D'_T, we consider $\{D_R, L_R\}$ as the training set, and query each instance $x^i_{T'}$ in D'_T using the Nearest Neighbor (NN) algorithm such that the label for $x^i_{T'}$ is equal to the label for x^j_R, which has the smallest Euclidean distance to $x^i_{T'}$, as given by

$$dist(x^j_R, x^i_{T'}) = \sqrt{\sum (x^j_R - x^i_{T'})^2} \tag{2}$$

D. Classifier Training

Because the obtained label set L'_T for D'_T is the same as for D_T, we take $\{D_T, L'_T\}$ as the training dataset and use it to train a classification algorithm for activity recognition in the view T.

III. Inter-Feature Mapping Problem

Given two datasets D_R and D_T with the same set of features f_S, an inter-feature mapping score for f^i_R and f^j_T is a real value ranging from 0 to 1 indicating the likelihood of f^j_T being associated with f^i_R.

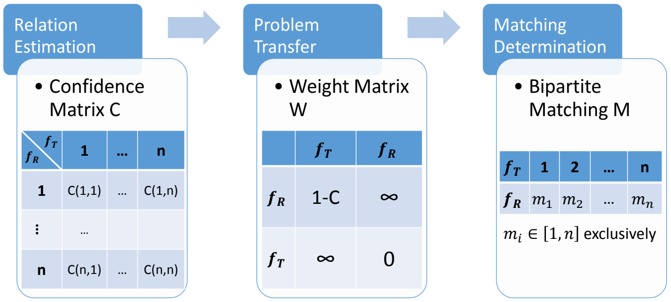
In an asynchronous scenario, the variable changes for f^i_R and f^j_T are independent and the sequence of the instances in the two views can be random. Therefore, we use Kullback-Leibler (K-L) divergence to estimate information loss by representing one feature using the other as below.

$$\Delta_{KL} = P(f^i_R) \times \log \frac{P(f^i_R)}{P(f^j_T)} \tag{3}$$

where P denotes the probability density function of the specific feature.

Given two datasets D_R and D_T, and a mapping score IFMS, *Inter-Feature Mapping* (IFM) is the problem of finding a perfect bipartite matching M between D_R and D_T, such that the overall IFMS of all the matched pairs is maximized.

Fig. 2 shows the workflow and the stepwise output of our inter-feature mapping algorithm.



- Confidence Matrix C

f_R \ f_T	1	…	n
1	C(1,1)	…	C(1,n)
⋮	…		
n	C(n,1)	…	C(n,n)

- Weight Matrix W

	f_T	f_R
f_R	1-C	∞
f_T	∞	0

- Bipartite Matching M

f_T	1	2	…	n
f_R	m_1	m_2	…	m_n

$m_i \in [1, n]$ exclusively

Fig. 2. Mapping Algorithm and Stepwise Output

Algorithm 1 Algorithm for Relation Estimation

```
Input: $D_R$ and $D_T$
$n \leftarrow$ number of features in $D_R$ and $D_T$
for $i$ is $1 \rightarrow n$ do
    normalize $f^i_R$, $f^i_T$ to $[0, 1]$
end for
for $bin$ in $BinSizes$ do
    for $i$ is $1 \rightarrow n$ do
        for $j$ is $1 \rightarrow n$ do
            $dist_{KL} = P(f^i_R) \times \log \frac{P(f^i_R)}{P(f^j_T)}$
        end for
        $f_{best} \leftarrow$ the feature in $D_T$ with smallest $dist_{KL}$
        $S^i_{bin} \leftarrow f_{best}$
    end for
end for
for $i, j$ is $1 \rightarrow n$ do
    $C(i, j) \leftarrow$ percentage of $j$ appears in $S^i$ for all $bins$
end for
Output: Confidence matrix $C$
```

A. Relation Estimation

IFMS is used to determine the quality of a mapping from one feature to another. Since K-L divergence is not symmetric, IFMS of two features is not symmetric either. In this algorithm, we only consider IFMS(f^i_R,f^j_T), which denotes how likely a feature in D_T can be mapped to a feature in D_R.

Algorithm 1 gives the pseudocode for the relation estimation algorithm. We first normalize all the variables. A number of bins with different sizes are used to acquire the probability density function for each feature. It results in a K-L divergence of (f^i_R, f^j_T) for every given bin value. The final output of the pairwise estimation is a matrix S. Each row in S represents a non-exclusive mapping from features in D_T to those in D_R for a given bin value, such that the i_{th} element in the row is the best match to f^i_R.

We convert matrix S to a confidence matrix C as shown in Fig. 2. The value computed for IFMS(f^i_R, f^j_T) is the same as $C(i, j)$, which represents the percentage of f^j_T happened to be the best match to f^i_R for all the given $bins$.

B. Problem Transfer

The confidence matrix C can be represented as a complete bipartite graph. We recall that a desired matching is a perfect matching on this complete bipartite graph that maximizes the sum of IFMS values. This problem is a classical matching problem in combinatorial optimization [4].

A *Feature Equivalent Graph* (FEG) is defined as follows. $FEG = (V,E)$, $V \in \{f_R, f_T\}$ is a complete graph where the edge weights are computed according to the following rules: (1) if $e(i,j)$ is an edge from f_R to f_T, then $w_{ij} = 1-C(i,j)$; (2) if $e(i,j)$ is an edge from f_T to f_R, then $w_{ij} = 0$; (3) if $e(i,j)$ connects two features in the same dataset, then $w_{ij} = \infty$

It can be observed that, the solution to a minimum bipartite matching is in fact a minimum cost Hamiltonian path on FEG. We transfer the inter-feature matching problem to a problem of detecting a minimum path that goes through all the vertices in the graph exactly once. This problem is known as Traveling Salesman Problem (TSP).

C. Matching Determination

We use a computationally simple heuristic algorithm, namely Nearest Neighbor, to solve the TSP problem. It is proved that the output of the Nearest Neighbor algorithm is within 25% of the Held-Karp lower bound [5], [6].

Given the weight matrix W, we start a path from a random feature in D_R, finding the unvisited neighbor with the smallest weight. Based on the definition of W, the selected feature must be in D_T. The path is gradually explored until all vertices are visited. Considering the output path may vary according to the exploration order of vertices, we run the path searching algorithm for n times. This results in n Hamiltonian paths. We then eliminate zero-weighted edges from f_T to f_R, but keep the edges from f_R to f_T. As a result, one Hamiltonian path will be computed as the perfect bipartite matching.

The n perfect bipartite matchings are first sorted according to the number of repetitions, and then sorted by the total cost of the corresponding path. The final matching M is the one appearing most frequently and having the smallest cost compared with other matchings with the same repetition.

IV. Validation

For evaluation purpose, we conducted an experiment where a participant was asked to attach a Shimmer [7] sensor firmly on the left wrist, and to perform several daily activities for 10 times each. The activities include 'normal walking' and 'running' at comfortable speed, and 'lifting up arms along medial-lateral (ML) direction'. Motion data were generated by 3D accelerometer and 3D gyroscope integrated in the Shimmer sensor platform. Data were transferred to a computer wirelessly over Bluetooth in real-time.

To simulate distinct sensor platform views, at least one platform parameter varies from one trial to another. Three types of programmable parameter were tested as listed in Table I. It is observed that any parameter change would cause significant difference in the sensor readouts even for the same activity.

TABLE I
Platform Parameter Space and Feature Space

Platform Parameter Space	
Parameters	**Values**
Sampling Frequency	51.2Hz, 102.4Hz
Accelerometer Sensitivity	±2g, ±4g, ±8g
Gyroscope Sensitivity	500dps, 1000dps
Feature Space	
min$_{\text{gz}}$, range$_{\text{gz}}$, rms$_{\text{gz}}$, rms$_{\text{ax}}$, rms$_{\text{ay}}$, max$_{\text{az}}$, min$_{\text{az}}$, mean$_{\text{az}}$, rms$_{\text{az}}$	

A. Data Collection

Several datasets were constructed exclusively according to specific assignment of the platform parameters in each trial. In a common situation, instead of directly using raw sensor data, some features were first extracted from sensor readouts, and then fed into data analysis algorithms [8]. For simplicity and generalization, we used nine statistical features, which were selected as highly predictive by running correlation-based feature selection (CFS) algorithm, to create dataset for evaluation. In Table I, 'gz' refers to z-axis signal from gyroscope, while 'ax', 'ay' and 'az' refer to x-, y- and z-axis acceleration signal, respectively.

B. Accuracy of Labeling Algorithm

To estimate accuracy of activity recognition over distinct sensor views, we directly compared labels assigned by our algorithms to the true labels manually noted during the experiment. Table II lists the average labeling accuracy and correct feature mapping ratio for three cases.

TABLE II
Average Labeling Accuracy

Parameter Changes	**Number of Test**	**Correct Feature Matching**	**Average Accuracy**
One	14	8.5/9	84.8%
Two	12	8.3/9	86.6%
Three	4	7/9	83.9%

For one parameter change, only one of the three above-mentioned parameters set differently between the reference dataset and the target. While, in the case of two and three parameter changes, a random combination of multiple parameters were different between the two datasets. Our algorithms achieved at least 83.9% accuracy in activity identification by taking the reference of a labeled asynchronous dataset.

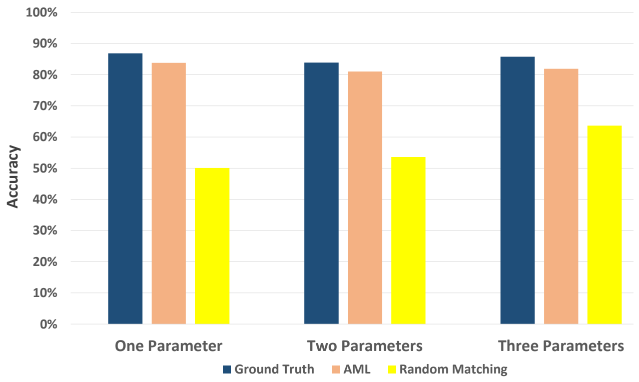


Fig. 3. Classification Accuracy Comparison for Three Cases

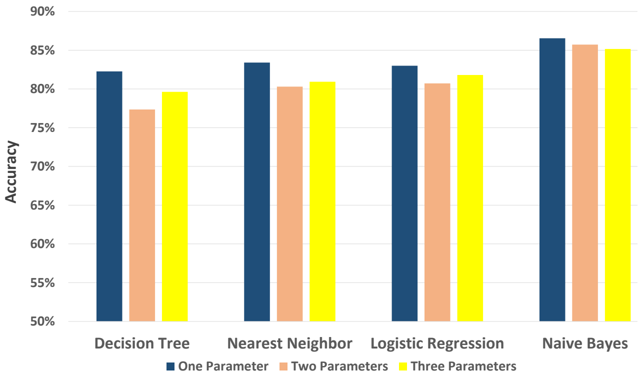


Fig. 4. Classification Accuracy of Various Machine Learning Algorithms

C. *Classification Accuracy*

To evaluate the effectiveness of activity classification using labeled dataset obtained by our algorithms, we first separated every dataset into two partitions with roughly equal size of instances for all three activities. We then randomly chose one subset as training, while the other was used for testing. We compared our results with the upper bound accuracy obtained using ground truth labeled data and a baseline labeling as the lower bound accuracy. The ground truth labeling refers to manually annotated activity-label in the experiment, and hence, is used to compute the upper bound accuracy. To estimate the baseline performance, we designed a random matching test case. It randomly generated an inter-feature mapping between the selected training set and another asynchronous dataset with true activity-label. Then, the training set was transformed and labeled following the same procedure described in Section II with the given matching result.

For classification on the separate test data, four popular machine learning algorithms, namely Decision Tree, Nearest Neighbor, Logistic Regression and Naive Bayes, were ran on Weka platform. Fig. 3 shows the results of average classification accuracy according to the number of different parameters between the training set and its reference asynchronous dataset for labeling. For ground truth labeling, it didn't use reference dataset, so the classification result was the same for a given training set and test set.

This comparison showed that classification accuracy based on AML labeled training set is 3.5% less than the upper bound for one parameter change, 3.39% less for two parameters change, and 4.51% less for three parameters change. However, given a randomized inter-feature matching, the accuracy dropped at least 25.8% over all the cases.

We also reported the classification accuracy of various machine learning algorithms leveraging with proposed AML labeling approach, as it shown in Fig. 4. It can be observed that, labeling the training set using AML algorithm and predicting the activity-label for test set using Nearest Neighbor can achieve the best performance, with an accuracy at least 85.2% for all the cases.

V. Conclusion and Future Work

In this paper, we introduced an Asynchronous Multi-view Learning (AML) approach for classification knowledge transfer. Our pilot study focused on activity recognition using wearable sensors. We demonstrated that we could develop an accurate feature mapping algorithm for knowledge transfer across asynchronous sensor views. The proposed approach enables the use of classification knowledge from an already configured platform to predict activity-label for asynchronously collected data on distinct platforms.

In this paper, we focused on knowledge transfer between wearable sensors for activity recognition. We are currently improving the robustness of our algorithms on more complex cases, as well as investigating the potential of AML approach for use with other health monitoring applications. Furthermore, we plan to study the AML problem for knowledge transfer across heterogeneous sensors.

References

[1] R. Fallahzadeh, M. Pedram, R. Saeedi, B. Sadeghi, M. Ong, and H. Ghasemzadeh, "Smart-cuff: A wearable bio-sensing platform with activity-sensitive information quality assessment for monitoring ankle edema," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops*.

[2] Y. Ma, R. Fallahzadeh, and H. Ghasemzadeh, "Toward robust and platform-agnostic gait analysis," in *IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*.

[3] D. Zakim and M. Schwab, "Data collection as a barrier to personalized medicine," *Trends in pharmacological sciences*, vol. 36, 2015.

[4] S. Khuller, S. G. Mitchell, and V. V. Vazirani, "On-line algorithms for weighted bipartite matching and stable marriages."

[5] C. Nilsson, "Heuristics for the traveling salesman problem."

[6] D. Johnson and L. McGeoch, "The traveling salesman problem: A case study in local optimization."

[7] A. Burns, B. Greene, M. McGrath, T. O'Shea, B. Kuris, S. Ayer, F. Stroiescu, and V. Cionca, "Shimmer - a wireless sensor platform for noninvasive biomedical research," *Sensors Journal, IEEE*, vol. 10, no. 9, pp. 1527–1534, Sept 2010.

[8] H. Banaee, M. U. Ahmed, and A. Loutfi, "Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges," *Sensors Journal, IEEE*, vol. 13, pp. 17 472 – 17 500, 2013.