Personalization without User Interruption: Boosting Activity Recognition in New Subjects Using Unlabeled Data

Ramin Fallahzadeh

School of Electrical Engineering & Computer Science
Washington State University
Pullman, WA 99164-2752
rfallahz@eecs.wsu.edu

Hassan Ghasemzadeh

School of Electrical Engineering & Computer Science
Washington State University
Pullman, WA 99164-2752
hassan@eecs.wsu.edu

ABSTRACT

Activity recognition systems are widely used in monitoring physical and physiological conditions as well as observing the short/long term behavioral patterns for the purpose of improving the health and wellbeing of the users. The major obstacle in widespread use of these systems is the need for collecting labeled data to train the activity recognition model. While a personalized model outperforms a user-independent model, collecting labels from every single user is burdensome and in some cases impractical. In this paper, we propose an uninformed cross-subject transfer learning algorithm that leverages the cross-user similarities by constructing a network-based feature-level representation of the data in source and target users and perform a best effort community detection to extract the core observations in target data. Our algorithm uses a heuristic classifier-based mapping to assign activity labels to the core observations. Finally, the output of labeling is conditionally fused with the prediction of the source-model to develop a boosted and personalized activity recognition algorithm. Our analysis on real-world data demonstrates the superiority of our approach. Our algorithm achieves over 87% accuracy on average which is 7% higher than the state-of-the-art approach.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**;
Transfer learning;

KEYWORDS

activity recognition; uninformed transfer learning; cross-subject boosting

ACM Reference format:

Ramin Fallahzadeh and Hassan Ghasemzadeh. 2017. Personalization without User Interruption: Boosting Activity Recognition in New Subjects Using Unlabeled Data. In *Proceedings of The 8th ACM/IEEE International Conference on Cyber-Physical Systems, Pittsburgh, PA USA, April 2017 (ICCCPS '17)*, 10 pages.

DOI: <http://dx.doi.org/10.1145/3055004.3055015>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICCCPS '17, Pittsburgh, PA USA

© 2017 ACM. 978-1-4503-4965-9/17/04...\$15.00

DOI: <http://dx.doi.org/10.1145/3055004.3055015>

1 INTRODUCTION

Advances in machine learning, signal processing, and low-power computation, in conjunction with the emergence of low-cost sensing/processing/communication hardware technologies have led to ever advancing pervasive computing and Internet-of-Things (IoT) applications. Wearable cyber-physical systems, as rapidly growing component of IoT, present numerous opportunities for real-time, context-aware and extensive health monitoring and intervention [2, 5, 14, 21]. The functionality of wearables is highly associated with the user's characteristics which warrants high adaptability in their computational model. These models are typically powered by supervised machine learning algorithms. In order for an adaptive model to maintain an acceptable performance, a large training dataset is needed which makes the training phase time consuming, labor intensive, and expensive. Data collection has been identified as a major obstacle in personalized and precision medicine [24]. Due to highly dynamic nature of wearable technologies, such a naive solution seems impractical. Examples of such variations are user-specific behavior, dynamic system architecture (addition of new sensors, replacing old sensors), various system platforms, change in the location/orientation of the sensors, etc. Collecting large enough training data to ensure adaptability to these variations is unrealistic given a large population of users are older adults. As a result, alternative solutions such as online/semi-supervised learning and transfer learning approaches are warranted [6, 22, 23].

The focus of our work is developing an autonomous transfer learning algorithm for cross-subject model adaptation. User-specific algorithms, unlike the user independent models, do not require a large training phase and tend to show higher performance [15]. The major drawback is the need for collecting a personalized training data which puts much burden on the user's shoulder. Online learning, semi-supervised learning, and supervised transfer learning techniques are examples of such approaches. Cross-subject unsupervised transfer learning, in contrast, leverages the similarity between the existing training data and the observations collected from the new user to automatically predict labels for the user-specific instances and adapt/re-train the model using the new observations and the predicted labels, in real-time and without supervision.

Transfer learning is a new research area. To the best of our knowledge, the need for a real-time unsupervised learning approach for personalized transfer learning in wearable activity recognition applications has remained unfulfilled to date. While most proposed studies rely solely on transferring raw data instances, feature-sets, and models, the relational knowledge transfer which utilizes the relations between source and target observations has not been explored by the community, as reported in a recent survey [6].

1.1 Transfer Learning in Activity Recognition

One of the abilities which makes human intelligence superior to artificial intelligence is its ability to understand the similarity between various contexts and extend the learned knowledge to new contexts. Transfer learning is the ability to extend what has been learned from one context to another nonidentical but similar context that shares common features. In the field of machine learning, transfer learning is defined as follows [18]:

Definition 1 (Transfer Learning). *Given a source domain D_s and learning task T_s , a target domain D_t and learning task T_t , transfer learning aims to help improve the learning of the target predictive function $f_t()$ in D_t using the knowledge in D_s and T_s , where $D_s \neq D_t$, or $T_s \neq T_t$.*

This definition can be easily be extended to the specific activity recognition domain where D_s and D_t are, respectively, the sensor observations made in source and target domain and can be represented in temporal, frequency, or feature space. The task T will develop a predictive function $f(\cdot)$ by learning a conditional probability distribution $P(y|x)$ where y and x are the activity labels and sensor observations.

Depending on how the source and target domains are defined, several variations of transfer learning can be considered. Such domain variation in wearable activity recognition can occur due to user differences, device or sensor type changes, skewed label distribution, etc. Transfer learning in activity recognition can also be categorized based on the availability of the labeled data in the source or target domains [6]. Where the ground-truth labels are (or not) available in the source domain, the transfer learning is often characterized as supervised (or unsupervised) transfer. In addition, if labels are (or not) available in target domain it is referred to as informed (or uninformed) transfer learning. The categorization is further illustrated in Fig. 1.

1.2 Related Work

Few aspects of transfer learning in activity recognition has been studied recently [6]. These studies can be categorized based on the target domain variation and availability of the labels, as described in Section 1.1. While the current work included various sensor modalities such as smart homes, wearables, and camera, our specific interest, in this paper, is the human centered and wearable activity recognition.

In addition to described categories, the current methodologies can be classified to synchronous or asynchronous transfer learning algorithms. While the majority of transfer learning algorithm are asynchronous, synchronous teacher/learner based methods have also been proposed, recently [20]. The biggest shortcoming of such approaches is the requirement of having an expert model present in real-time which is not usually affordable. In addition, the training of the expert requires enormous training efforts. However, in multi-node wearable system such algorithms are shown to be effective [19, 20].

The algorithm proposed in this paper aims to perform uninformed and supervised cross-subject transfer learning. In other words, we have a labeled source dataset acquired from the source

Figure 1: The categorization of transfer learning in activity recognition based on and the source of source-target variation and availability of activity labels in source and target data.

subject and an unlabeled target dataset. The goal is to train a personalized and accurate activity recognition algorithm in the target domain that can boost the recognition performance in unseen users (with skewed feature space). To the best of our knowledge, this area of research is almost unexplored to date. The closest prior research conducted is the informed supervised/unsupervised learning where the assumption is that a partial labeled data is available in the target domain [1, 9]. This area of research sometimes manifests in active learning research [3, 13].

Few studies [7, 25] suggested using unsupervised clustering algorithms such as Kmeans clustering algorithm to update the model trained on the source dataset. The biggest shortcoming of such approach is the naive assumption that the number of activities (i.e., the desired number of clusters) is a known parameter in target dataset. In contrast, in our approach we only define a lower bound on the size of clusters and propose a greedy solution to find the most desired partitioning. We demonstrate that naive clustering (and naive fusion of base-classifier and clustering algorithm) performance is not significant in a more realistic and complex scenarios where the a larger number of activities are present, as shown in more details in our analysis in Section 4.2.

1.3 Contribution

In this paper, we introduce the concept of relational knowledge transfer for human-centered activity recognition applications. Our framework uses a novel knowledge transfer learning algorithm to discover the relationships between source and target domains in order to asynchronously extract sufficient knowledge to develop its own personalized machine learning algorithm. Developing the customized activity recognition algorithm is based on the observations made in real-time in target domain, adaptively fused with the predictions of the source model. Our methodology dismisses

the implicit assumption that the data gathered in human-centered applications are from the same domain and probability distribution and instead aims to optimally rely on its own observations and their relation to source observations in order to build and update an accurate machine learning algorithm. The process is conducted in real-time without need for a synchronous teacher model [20] or supervised labeling effort to generate a new training dataset.

As mentioned, our target application, in this paper, is the human activity recognition using wearable sensors. We aim to address the problem of reusing an activity recognition system developed and trained on a single (or multiple) user(s) on a new user without any training efforts while maintaining the accuracy of the system. In this scenario, the labeled observations from the first group form the source view/domain and the unlabeled observations made by the new user form the target view/domain. Such a learning approach is sometimes referred to as multi-view learning [20]. We introduce a network of highly similar observations and perform greedy community detection for robust partitioning and detection of valuable observations in target domain. We formulate the problem of optimized target dataset mapping using linear programming and a uneven bipartite graph model and solve it using a novel adaptive decision fusion. We use real-word wearable sensor data to evaluate our model in comparison against our baselines, upper-bound, and state-of-the-art solutions such as classification via clustering, bagging and random forest algorithms.

2 PROBLEM STATEMENT

Fig. 2 shows the evolution of a wearable activity recognition system as it is adopted by new users. Initially, a machine learning model (i.e., activity recognition algorithm) is trained based on the annotated data (source dataset) collected from one or multiple train subjects (Fig. 2a). The training phase is labor intensive as a large dataset is required to improve generalization of the model. In Fig. 2b, a new user adopts the system. Due to complexity and high variation of daily activities, the existing model exhibits significant drop in accuracy. The results of such cross-subject performance decline is discussed in Section 4.1. As the user starts using the system, an unlabeled user-specific dataset is gradually collected (target dataset). We aim to utilize the collective knowledge extracted from the source and target datasets to predict activity labels for target dataset and train a user-specific model that ensures high accuracy, as shown in Fig. 2c.

2.1 Problem Definition

Let X be a sequence of observations made from one sensing node containing a finite set of sensor. The observations made within a single time segment are represented in a K -dimensional feature space, where K is the number of features. As a result, vector $X_i = \{f_{i1}, f_{i2}, \dots, f_{iK}\}$ represents physical attributes during the time segment i . The set of all physical activity labels, with the size of m , forms a label space $L = \{l_1, l_2, \dots, l_m\}$. The activity recognition algorithm assigns an activity label $l \in L$ to the observation X_i using a predictive function that is a conditional probability distribution $P(A|X_i)$ trained based on the labeled training dataset. The accuracy of such model is guaranteed as the future observations and activities will belong to the same distribution. Therefore, the

activity recognition model trained on specific user(s) will be able to accurately attribute the labels to future observations. However, in a more realistic scenario, once a model is trained on train subjects, it will be available to new users. The observations made by the new user will not follow the same conditional probability distribution as the distribution of features and labels are expected to be similar to the training dataset however not identical. For instance, given the complex nature of physical movements, different individuals do not perform a particular activity in exactly the same fashion to the extent that the way that certain physical activities are performed is being used by researchers for user identification [10]. Without loss of generality, we consider a single node activity recognition system, however, our methodology can be easily extended to multi-node body sensor networks.

Problem 1 (Cross-Subject Transfer Learning). *Let DS_s be the labeled dataset collected from source subject(s) and $DS_t = \{X_{t1}, X_{t2}, \dots, X_{tN}\}$ be the accumulative dataset composed of N unlabeled observations made from the new subject. Furthermore, let L_t be the activities performed by the new user. The Cross-Subject Transfer Learning (CSTL) is the problem of assigning a label $l \in L$ such that the error of label assignment is minimized. Once enough observations are labeled with activities, training a personalized activity recognition is straightforward.*

The minimization process aims to minimize the miss-assignments within the L_t set. In the rest of this section, we further formulate the problem and overview the proposed solution.

2.2 Problem Formulation

The Cross-Subject Transfer Learning problem defined in Section 2.1 is formulated using linear programming as follows.

$$\text{Minimize } \sum_{ij} x_{ij} \epsilon_{ij} \quad (1)$$

Subject to

$$\sum_{1 \leq j \leq |L|} x_{ij} = 1, \forall i \in \{1, \dots, N\} \quad (2)$$

$$\delta_j \leq \sum_{1 \leq i \leq N} x_{ij} \leq \Delta_j, \forall j \in \{1, \dots, |L|\} \quad (3)$$

$$x_{ij} = \begin{cases} 1, & \text{if label } l_j \text{ is assigned to observation } X_i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where ϵ_{ij} is the error of assigning label l_j to observation X_i . Equation (1) aims to minimize the overall error of label assignment in the target domain. The binary variable x_{ij} is set to '1' if X_i is assigned to l_j and is set to '0' otherwise. Constraint (2) guarantees that for each target observation, exactly one activity label is assigned. Finally, constraint (3) ensures that the size of the data cluster labeled by an activity j satisfies the lower-bound δ_j and the upper-bound Δ_j . The main challenge is that unlike the source domain where the ground truth labels are available, in the target domain without ground truth labels, ϵ_{ij} is unknown to us.

2.3 Solution Overview

In our knowledge transfer based approach, the source domain acts as an offline teacher and the target view acts as a real-time learner.

Figure 2: The overview of cross-subject activity recognition. A wearable activity recognition system (consisted of one or multiple nodes) uses a model generated based on annotated data, i.e., source view (a). A new user adopts the system. An activity learning module is in charge of constructing a user-specific dataset in real-time, i.e., target view (b). When learning is completed, the new user starts using the updated model (c).

The source view transmits a training dataset containing features and its corresponding ground truth labels. We refer to this as the source dataset (DS_s). The source dataset contains sufficient observations of each activity label. It is, in fact, the same training dataset used to train the activity recognition model for the source subject. Therefore, the only ground truth that our model uses is the training data previously gathered to develop the existing model in the source view.

Once the new user starts using the system, the target view begins collecting its own observations in real-time that are initially without labels and, therefore, cannot be used for training a model. We refer to the collected data, in this step, as the target dataset (DS_t). After a sufficient amount of data is collected, our activity learner module extracts feature-based similarities between the source and target datasets and executes an intelligent label transfer algorithm to construct a labeled target dataset. Once we have the labeled target dataset, developing a personalized activity learning model for the new subject is a straightforward task.

Fig. 3 shows the steps taken inside the learning module, in more details. The inputs are the source and target datasets. The source dataset (DS_s) is used to develop a supervised activity recognition model (i.e., source model). While various supervised algorithms can be considered, some techniques are more robust against over-fitting, meaning that the developed model is less biased toward the training data. These algorithms are often based on random ensembles of features (e.g., random forest algorithm) or instances (e.g., bagging algorithm). The source model is tested on the target dataset to assign supervised label predictions, that are later used in our label fusion algorithm. The target dataset (DS_t) is used for constructing a network of highly similar observations in feature space. The observations that have a correlation higher than a preset threshold will be connected in our network. The threshold is set by a greedy algorithm, described in Section 3.1. From the network, the highly similar observations are extracted as communities representing a particular activity label. The *minimum error subset mapping* module,

then, will heuristically map the communities to a subset in our label space (see Section 3). At last, the labels acquired from both source and target views will be conditionally fused, based on their prior per-label performance (see Section 3.3).

Algorithm 1 summarizes the main steps in our proposed knowledge fusion transfer learning approach.

Algorithm 1 Knowledge-fusion-based Cross-Subject Transfer Learning

- 1: *Collect target observations;*
 - 2: *Construct the similarity network (SN_t) using the DS_t ;*
 - 3: *Perform best effort community detection on SN_t to identify communities corresponding to a subset of labels in the label space;*
 - 4: *Perform Minimum-Error Subset Labeling using collected DS_t and DS_s and construct a labeled subset of DS_t ;*
 - 5: *Construct a supervised source model and predict labels for $\forall X \in DS_t$;*
 - 6: *Given the conditional probability of labels assigned from both step 4 and step 5, being a correct label, assign the most probable label (weighted label fusion);*
 - 7: **If** not enough observations per label were extracted in step 4 (it can be observed from the convergence of community mapping results);
 - 8: *Go to step 1;*
 - 9: *Train a personalized activity recognition model using the constructed labeled DS_t ;*
-

3 BEST EFFORT UNSUPERVISED LABELING

We formally define the problem of ‘*minimum error subset labeling*’ and propose a greedy heuristic solution. The objective is to use the relational knowledge transferred from the labeled source dataset, instead of relying on the partially biased source model, to predict activity labels for unlabeled target observations and construct a labeled target sub-dataset of DS_t , which can be fed into the label

Figure 3: The block diagram of our learning module. It includes network construction, community detection, minimum error subset mapping, and a weighted label fusion of unsupervised and supervised label mappings.

fusion module. The key boosting function is to have such personalized instances fused with the source model to boost the activity recognition in the target view.

We note that assigning a label to every single observation in DS_t , using an unsupervised method, is less likely to achieve a high accuracy, especially in presents of a large and complex set of activities (i.e., a more realistic settings). Instead, we aim to find the observations that can be labeled with higher confidence. As a result the constructed labeled dataset in target view is a more accurate assignment, containing only highly confident instances, resulting in a more accurate mapping. The challenging task is to find such an accurate (*Observation Subset, Activity Label*) mapping. We refer to such observations as core observations.

Definition 2 (Core Observation). *An observation X_i is called a core observation for a label $l \in L$ if and only if it has the highest correlation with the source observations with the same label, with respect to all non core observations of l such that there are at least δ_j and at most Δ_j of such core observations.*

Problem 2 (Minimum-Error Subset Labeling). *Given a source dataset (DS_s) and a target dataset (DS_t), find a subset of DS_t (core observations) that can be labeled with activity labels L_t such that the error of labeling is minimized.*

In order to simplify the problem, we first group the observations in both source and target view into several partitions. In such data representation, highly similar observations are represented as one cluster. In source view, the clustering task is trivial, as the ground truth labels are available. Therefore, we place the instances that share the same label in one cluster. Therefore, the number of clusters equal the number of activities, as follows:

$$DS_s = C_1^s \cup C_2^s \cup \dots \cup C_{|L|}^s \quad (5)$$

such that

$$C_1^s \cap C_2^s \cap \dots \cap C_{|L|}^s = \emptyset \quad (6)$$

3.1 Network-based Partitioning

In target view, unlike the source view, the ground truth labels are unknown. In addition, since gathering the target data is a collective process, the exact number of activities performed by the target user is unknown. That is the reason why we cannot use a clustering algorithm such as Kmeans. Because of the resoluteness that network-based community detection clustering algorithms provide, we borrow the network based concepts in this work to achieve higher clustering performance (as shown in our analysis in Section 4.2). We measure similarity between sensor observations (i.e., the

nodes in the similarity network) in feature space to detect communities of similar nodes. When partitioning the target view, we make the following realistic assumptions: (1) unlike the source dataset, the target dataset is collected in an open environment situation and in presence of less controlled instances and transitional moments between activities, making some of the observations non desirable in our final training dataset; (2) due to the high complexity and similarity of several activity of daily living and their uneven distribution, naive clustering results will potentially output fuzzy and/or unbalanced clusters of observations.

We use the training data from the source subjects to find the optimal similarity threshold that maximizes the clustering performance. The similarity score used in our approach is the *normalized cosine similarity* due to its superior performance in our high-dimensional feature space. As stated in Equation (3), we also define an lower-bound on the size of clusters to discard the small communities that represent the outliers, resulting in a subset of instances in DS_t , potentially the core observations. Finding the optimal similarity threshold (α) can be formulated as follows:

$$\text{Maximize } \alpha \quad (7)$$

such that

$$|C| = \{c | \delta \leq |c| \leq \Delta\} = |L| \quad (8)$$

Where C is the set of all communities (clusters) that satisfy the size constrains in Equation (3). The greedy algorithm for finding α is described in Algorithm 2. We note that optimizing the threshold is a part of training phase and is performed on the source users. The average value is, then, used for partitioning the target dataset as explained in Algorithm 3.

Algorithm 2 The greedy algorithm for finding the optimal similarity threshold (α)

- 1: **For** each source user U_i :
- 2: $\alpha \leftarrow 0.500$; ▷ Initially set to a small value;
- 3: Construct the similarity network ($SN_{U_i}^\alpha$) using $DS_s^{U_i}$;
- 4: Perform community detection on $SN_{U_i}^\alpha$;
- 5: Discard outlier communities, resulting the set C ;
- 6: **If** $|C| < |L|$: $\alpha \leftarrow \alpha + 0.005$; **Go to line 3**;
- 7: **If** $|C| == |L|$ **OR** $\alpha == 1$; **Continue**;
- 8: **Return** Average α over all source users;

Having extracted the valid communities, we propose a graph model to solve the problem of finding the final core clusters and assigning labels in polynomial time.

Algorithm 3 Network construction and community-based dataset partitioning

```

1: procedure CONSTRUCT SIMILARITY NETWORK( $DS, \alpha$ )
2:   Create a node for each  $X \in DS$ ;
3:   For each pair of  $X_i, X_j \in DS$ :
4:     If  $NormalizedCosineSim(X_i, X_j) \geq \alpha$ :
5:       Connect the nodes corresponding to  $X_i$  and  $X_j$ ;
6:   Return  $SN_{DS}^\alpha$ ;
7: procedure DATASET PARTITIONING( $SN, \delta$ )
8:    $C \leftarrow$  Execute a community detection algorithm;
9:    $\forall c_i \in C$  if  $|c_i| < \delta$  remove  $c_i$ ;
10:  Return  $C$ ; ▷ The set of valid clusters;
    
```

3.2 Weighted Labeling Graph

We propose a complete bipartite graph model using the identified clusters in source and target view and their pairwise degree of similarity. To measure similarity, we again use the normalized cosine similarity between clusters in both views, having observation instances represented in feature space. The significance of our graph model is that it can be used to generalize Problem 2 to a bipartite matching problem.

Definition 3 (Weighted Complete Similarity Graph). *Let C^s be the set of clusters in source view where each cluster is associated with an activity label and C^t be the set of unlabeled clusters (i.e., valid communities) in target view. $G(V, E, W)$ is a weighted complete bipartite graph where $V = \{C^s \cup C^t\}$ and E and W , respectively, refer to the set of edges and their corresponding weights connecting the vertices in C^s to the vertices in C^t .*

The weight $\omega_{ij} \in W$, in our weighted complete similarity graph, represents the degree of similarity between each $(C^s - C^t)$ pair and is given by

$$W = \{\omega_{ij} = \text{norm_cosine_sim}(C_j^s, C_i^t)\} \quad (9)$$

in the case where there are multiple source users available, the set W will be the average computed weight using all source user, separately.

Fig. 4 illustrates an example of the construction of our graph model. Fig. 4a shows the unlabeled target observation in a two dimensional feature space. The labels are initially unknown. In Fig. 4b, our unsupervised community-based algorithm partitions the target dataset with respect to the predefined lower-bound on the cluster size. We note that in this step, the number of extracted clusters could be less than the size of ground-truth labels ($|C^t| < |L_t|$). We create the weighted complete bipartite graph using the clusters ids in target view and the clusters ids (i.e., the available activity labels) in source view, as shown in Fig. 4c. The weights on each edge is computed from Equation (9). We note where $|C^s| \neq |C^t|$, we add dummy nodes to create a complete bipartite graph.

We propose a simple polynomial time label transfer algorithm that maps the activity labels from $|C^s|$, to a subset of clusters in $|C^t|$. We want maximum similarity in our bipartite matching. Using the constructed weighted complete bipartite model, our algorithm uses maximum weight bipartite matching to perform the maximum similarity label mapping, as shown in Fig. 4d. We use Hungarian

Algorithm ¹ (a widely used weighted bipartite matching algorithm with $O(n^3)$ time complexity). The edges incident to dummy nodes have weights equal to zero. As a result, the nodes matched to the dummy nodes are excluded after the matching. As a result, some of the labels in label space L . It could happen because of two reasons: (1) one or more labels in L simply were not performed by the target user, at least so far; (2) our network based clustering algorithm was not able to identify those labels as valid clusters, therefore we call it a best effort algorithm which may sacrifice some labels for higher clustering accuracy of those that were identified. Finally, the assigned labels are propagated to the target observations inside each cluster, as shown in Fig. 4e. The main steps in our unsupervised label transfer algorithm is further described in Algorithm 4.

Algorithm 4 Best effort unsupervised labeling

INPUT: labeled source DS_s and unlabeled target DS_t

RESULT: partially labeled target dataset $DS_t^{Labeled} \subset DS_t$

```

1:  $C^s \leftarrow$  Partition  $DS_s$  using given  $l \in L$ ;
2:  $C^t \leftarrow$  Perform best effort clustering on  $DS_t$ ;
3: Create  $G(V, E, W)$  from  $C^s$  and  $C^t$ ;
4: Perform maximum weight bipartite matching (Hungarian Alg.);
5: Propagate the labels and construct  $DS_t^{Labeled}$ ;
    
```

3.3 Adaptive Label Fusion

Once we have the partially labeled dataset ($DS_t^{Labeled}$) from Algorithm 4, we also construct a state-of-the-art robust supervised machine learning algorithm (e.g., random forest, bagging, random tree, etc.), called source model, trained on the source user(s) data to provide predictions on target observations. In such setting, for each observation, we have at least one prediction from the source model and a possible mapping from the $DS_t^{Labeled}$. Two approaches can be taken with this information: (1) naive label fusion, and (2) adaptive or conditional label fusion.

3.3.1 Naive label fusion. In this approach, we assume the accuracy of both supervised and unsupervised labeling algorithms are user-independent and therefore the prior assumption is that the output label of the algorithm is unconditionally true. Therefore the prior probability distribution of the output label $l' \in L$ is defined as follows:

$$P(l|l') = \begin{cases} 1 & l' = l \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where l and l' , respectively, denote the ground-truth label and the assigned label by either the unsupervised or the supervised algorithms.

Under such a circumstance, for each observation in DS_t , a majority vote will assign the final label. Majority voting is a widely used approach that is being used, for instance, in random forest algorithm to conclude the predicted label. When the labels assigned by our label transfer algorithm and source model disagree (the final vote is a tie), one label can be randomly chosen or the observation instance can be ruled out as a *uncertain observation* and discarded from the final labeled dataset.

¹https://en.wikipedia.org/wiki/Hungarian_algorithm

Figure 4: An example showing the graph model construction and unsupervised transferring of the labels

3.3.2 Conditional label fusion. Since both algorithms can be easily tested on the source users dataset, a prior knowledge of their performance can be learned in a form of a conditional probability distribution: $P(L|L')$. Therefore, the predictive function can be reformulated as:

$$f(X_i) = P(L|L')P(L'|X_i) \quad (11)$$

A similar technique has been used in adaptive boosting² where prior knowledge on the performance of a learning algorithm is utilized iteratively for emphasizing on learning more difficult data instances. The difference of such adaptive label fusion, compared to naive fusion, is that when we have a tie in label assignments, the label with higher predictive value will be chosen which can further boost the labeling accuracy.

4 EXPERIMENTAL RESULT

To demonstrate the effectiveness of the proposed framework, we use the real-world UCI ‘Daily and Sports Activities Data Set’ [4]. A 3D motion tracker placed on torso was used to collect triaxial acceleration data at 25Hz sampling rate. Eight subjects were asked to perform 15 activities (such as sitting, walking, running at different speed, cycling, rowing, etc.), each for duration of five minutes. Each trial was divided into five-second segments of activity where 20 statistical and morphological features were extracted from each segment. Refer to [11] for details on the feature set utilized in the experiments. The data set contained over 900,000 samples of acceleration data.

4.1 Cross-Subject Performance Evaluation

In our initial analysis, we focused on the activity recognition performance on the dataset. Several popular machine learning algorithms were initially considered and decision tree algorithm was chosen because of its superior performance. We evaluated the performance using three evaluation methods, as shown in Fig. 5. In our first evaluation, we trained a model based on the data from one subject and tested the model on the data from a different subject (i.e., one to one validation). For each subject, we trained seven distinct models using the other subjects and reported the average classification accuracy. In the second evaluation, we trained a user-independent model using seven subjects and tested on one unseen subject (i.e.,

Figure 5: Activity recognition accuracy using various evaluation methods.

leave one subject out validation). Finally, a typical 10-fold cross validation on the entire dataset containing all subjects was performed. The results confirms our hypothesis of a significant decline with unseen data from a new subject. One to one model showed an over 61% performance decline in comparison to the 10-fold cross validation, on average. This number was 36% for leave one subject out model, where comparatively a much larger training dataset was used. These results show the importance of having a cross-subject learner to maintain high performance with a new user.

4.2 Labeling Accuracy

In this part of the analysis, we evaluated the accuracy of labeling the new user dataset (i.e., target view) using our approach, explained in Algorithm 4. Our network-based clustering approach is novel. In absence of existing methods, we devised several baselines. Our baseline approaches use popular unsupervised clustering algorithms to partition the target dataset and, similar to the proposed approach, use a similarity/distance measure to map the labels in source view to the generated clusters. The average mapping accuracy for each subject is illustrated in Fig. 6. The Kmeans algorithm requires the desired number of output clusters as an input (set to 15 in our experiments), which limits its practicality. In contrast, The DBSCAN clustering and Expectation Maximization (EM) methods automatically optimize the number of output clusters. The Xmeans

²<https://en.wikipedia.org/wiki/AdaBoost>

Figure 6: Label mapping accuracy using the proposed network-based clustering vs various clustering baselines.

Table 1: Partitioning the target dataset using our best effort community detection

-	# of non-mapped instances	Dataset utilization (%)	# of valid communities
User 1	182	76.2	15
User 2	88	88.5	14
User 3	255	66.7	15
User 4	96	87.4	11
User 5	46	94.0	11
User 6	79	89.7	15
User 7	257	66.4	13
User 8	167	78.2	15

algorithm (an extension of Kmeans method) only requires a lower-bound and an upper-bound on the number of desired clusters to find the optimal partitioning.

As suggested by Fig. 6, while the proposed network-based algorithm achieves a significantly high accuracy (84% on average), the naive partition-map approaches fail to achieve desirable performance, mainly due to the low performance of naive clustering in correctly partitioning a large and complex set of physical activity instances. For instance, DBSCAN and EM methods only detected an average of 6 and 13 clusters, respectively. This number is far less than the number of actual activity labels in this data set. After extracting the valid clusters, we mapped them to the activity labels using our bipartite graph model and Algorithm 4. We note that several network community detection algorithms were analyzed and the *Fast Greedy Algorithm* [17] was picked because of its low complexity and significant performance. The X axis in Fig. 6 indicates the excluded user (target subject) in each analysis. For constructing the similarity network, a threshold $\alpha = 0.99$ was used after executing Algorithm 2. Table 1 lists the details of our target view partitioning. It can be observed that our partitioning algorithm while having a significant mapping accuracy is able to map labels to a large portion of the target dataset (over 80% utilization on average).

Figure 7: Activity recognition by training various machine learning algorithms on the partially labeled DS_t .

4.3 Comparative Performance Evaluation

In the last part of our analysis, we compare the performance of our cross-subject transfer learning algorithm described in Algorithm 1 using different label fusion techniques, from Section 3.3. First, we used only the partially labeled target dataset generated by our clustering solution and tested it on the same user dataset with ground-truth labels. For the choice of machine learning algorithm, we selected decision tree, bagging, and random forest on Weka machine learning library [12]. Random forest and bagging algorithms are designed to show robustness against minor variations (such as user variation, in our case). Their accuracy of activity recognition is shown in Fig. 7. Due to superior performance of random forest algorithm, it is used as the base supervised algorithm for further analysis. We used 10-fold cross validation to ensure the test observations are not used for training of the same model.

We considered three variations of the proposed model based on the choice of label fusion: (1) naive fusion with discarding the instances with tied mapping decisions; (2) naive fusion with randomly picking a label when a tie occurs; and (3) adaptive fusion based on the prior knowledge of mapping accuracy (evaluated on source users), as previously explained in Section 3.3. The source model, in all cases, is a random forest algorithm trained on the source users. The results are illustrated in Fig. 8. As expected, adaptive fusion achieves higher accuracy with 8% – 10% improvement in activity recognition accuracy.

We, further, compared the final activity recognition performance of the proposed algorithm with adaptive fusion with our baselines (only the source model), state-of-the-art algorithms, and experimental upper-bound. In baseline approaches, we used a model (bagging and random forest algorithms) trained on a source subjects to classify the observation in target dataset. The state-of-the-art transfer learning models are classification via clustering algorithm (CvC) [16], and classification via regression (CvR) [8]. These approaches use clustering and regression on the test dataset to boost the robustness against data variations. Furthermore, we report the experimental upper-bound that is the classification accuracy of a 10-fold cross validation of the model trained on the ground-truth labels. The average performance per subject is shown in Fig. 9. These results clearly demonstrate the significantly superior performance of the proposed algorithm. The average accuracy improvement is between 21% (against CvC) to 7% (against random forest algorithm).

Figure 8: Activity recognition of the proposed transfer learning algorithm after naive fusion with discarding ties, naive fusion with randomly picking ties, and adaptive fusion.

Figure 9: Activity recognition accuracy using the proposed algorithm vs baselines and experimental upper-bound.

5 DISCUSSION AND FUTURE DIRECTION

The superiority of the proposed methodology relies on the fact that it knows the between-subject variations in feature space and takes advantage of this knowledge to not only identify core observations that are better representatives of the activities performed by the user, but also intelligently decide whether or not use the unsupervised mapping algorithm or the user-independent source model. As a result, the constructed model in this way is less biased toward the source observations. In other words, our model utilizes the cross-subject relational knowledge provided by the source view rather than merely relying on the source observations and the model created on those instances.

Due to space limitations, in our analysis, we only considered the case where the activities performed in target view and source view are identical. However, in a more realistic scenario, the set of activities performed by a particular subject (i.e., L_t) may be a subset of the set of pre-defined activities (L), specially in early stages of data collection. We aim to extend our case study to also addressing the small target dataset problem in our future work.

6 CONCLUSIONS

Upon rapid growth in wearable technologies, the underlying machine learning algorithms need to move toward higher specificity to increase user adaptability. While traditional machine learning uses a limited set of training data to develop a generalized model, such a subject-independent model is unable to maintain the initial

training accuracy. In this paper, we proposed a novel autonomous cross-subject transfer learning approach that utilizes the feature-level correlations between the source and target user data and constructs a high performance personalized model by combining the knowledge from labeled source data and unlabeled target data. Our analysis on real-world data demonstrated the superiority of our approach against several state-of-the-art solutions including classification via clustering/regression, bagging, and random forest algorithms with 21% – 7% increase in accuracy. Our transfer learning algorithm achieved over 87% classification accuracy, on average.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was supported in part by the National Science Foundation under Grant No.: 1566359. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

- [1] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. 2012. StreamAR: incremental and active learning with evolving sensory data for activity recognition. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, Vol. 1. IEEE, 1163–1170.
- [2] Parastoo Alinia, Ramyar Saeedi, Ramin Fallahzadeh, Ali Rokni, and Hassan Ghasemzadeh. 2016. A Reliable and Reconfigurable Signal Processing Framework for Estimation of Metabolic Equivalent of Task in Wearable Sensors. *IEEE Journal of Selected Topics in Signal Processing* (2016).
- [3] Salikh Bagaveyev and Diane J Cook. 2014. Designing and evaluating active learning methods for activity recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 469–478.
- [4] Billur Barshan and Murat Cihan Yüsek. 2014. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *Comput. J.* 57, 11 (2014), 1649–1667.
- [5] Marie Chan, Daniel Estève, Jean-Yves Fourniols, Christophe Escriba, and Eric Campo. 2012. Smart wearable systems: Current status and future challenges. *Artificial intelligence in medicine* 56, 3 (2012), 137–156.
- [6] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems* 36, 3 (2013), 537–556.
- [7] Wan-Yu Deng, Qing-Hua Zheng, and Zhong-Min Wang. 2014. Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Networks* 53 (2014).
- [8] E. Frank, Y. Wang, S. Inglis, G. Holmes, and I.H. Witten. 1998. Using model trees for classification. *Machine Learning* 32, 1 (1998), 63–76.
- [9] Enrique Garcia-Ceja and Ramon Brena. 2015. Building Personalized Activity Recognition Models with Scarce Labeled Data Based on Class Similarities. In *International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer, 265–276.
- [10] Jürgen T Geiger, Maximilian Kneißl, Björn W Schuller, and Gerhard Rigoll. 2014. Acoustic gait-based person identification using hidden Markov models. In *Proceedings of the 2014 Workshop on Mapping Personality Traits Challenge and Workshop*. ACM, 25–30.
- [11] Hassan Ghasemzadeh, Navid Amini, Ramyar Saeedi, and Majid Sarrafzadeh. 2015. Power-aware computing in wearable sensor networks: An optimal feature selection. *IEEE Transactions on Mobile Computing* 14, 4 (2015).
- [12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
- [13] HM Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2016. Active learning enabled activity recognition. *Pervasive and Mobile Computing* (2016).
- [14] M Shamim Hossain and Ghulam Muhammad. 2016. Cloud-assisted industrial internet of things (iiot)-enabled framework for health monitoring. *Computer Networks* 101 (2016), 192–202.
- [15] Heli Koskimäki and Pekka Siirtola. 2016. Adaptive model fusion for wearable sensors based human activity recognition. In *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 1709–1713.

- [16] Manuel Ignacio Lopez, JM Luna, C Romero, and S Ventura. 2012. Classification via Clustering for Predicting Final Marks Based on Student Participation in Forums. *International Educational Data Mining Society* (2012).
- [17] Mark EJ Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical review E* 69, 6 (2004), 066133.
- [18] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.
- [19] Seyed Ali Rokni and Hassan Ghasemzadeh. 2016. Autonomous sensor-context learning in dynamic human-centered internet-of-things environments. In *Proceedings of the 35th International Conference on Computer-Aided Design*. ACM, 75.
- [20] Seyed Ali Rokni and Hassan Ghasemzadeh. 2016. Plug-n-learn: automatic learning of computational algorithms in human-centered internet-of-things applications. In *Proceedings of the 53rd Annual Design Automation Conference*. ACM, 139.
- [21] Ramyar Saeedi, Navid Amini, and Hassan Ghasemzadeh. 2014. Patient-Centric On-Body Sensor Localization in Smart Health Systems. In *The Asilomar Conference on Signals, Systems, and Computers*. IEEE.
- [22] Ramyar Saeedi, Hassan Ghasemzadeh, and Assefaw Gebremedhin. 2016. Transfer Learning Algorithms for Autonomous Reconfiguration of Wearable Systems. In *IEEE International Conference on Big Data*. IEEE.
- [23] Lina Yao, Feiping Nie, Quan Z Sheng, Tao Gu, Xue Li, and Sen Wang. 2016. Learning from less for better: semi-supervised activity recognition via shared structure discovery. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 13–24.
- [24] David Zakim and Matthias Schwab. 2015. Data collection as a barrier to personalized medicine. *Trends in pharmacological sciences* 36, 2 (2015), 68–71.
- [25] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. 2011. Cross-people mobile-phone based activity recognition. In *IJCAI*, Vol. 11. Citeseer, 2545–2550.