Plug-n-Learn: Automatic Learning of Computational Algorithms in Human-Centered Internet-of-Things Applications

Seyed Ali Rokni, and Hassan Ghasemzadeh
Embedded and Pervasive Systems Laboratory (EPSL)
School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164–2752
{alirokni, hassan}@eecs.wsu.edu

ABSTRACT

Wearable technologies play a central role in human-centered Internet-of-Things applications. Wearables leverage computational and machine learning algorithms to detect events of interest such as physical activities and medical complications. A major obstacle in large-scale utilization of current wearables is that their computational algorithms need to be re-built from scratch upon any changes in the configuration of the network. Retraining of these algorithms requires significant amount of labeled training data, a process that is labor-intensive, time-consuming, and infeasible. We propose an approach for automatic retraining of the machine learning algorithms in real-time without need for any labeled training data. We measure the inherent correlation between observations made by an old sensor view for which trained algorithms exist and the new sensor view for which an algorithm needs to be developed. By applying our real-time multi-view autonomous learning approach, we achieve an accuracy of 80.66% in activity recognition, which is an improvement of 15.96% in the accuracy due to the automatic labeling of the data in the new sensor node. This performance is only 7.96% lower than the experimental upper bound where labeled training data are collected with the new sensor.

1. INTRODUCTION

Internet of Things (IoT) is emerging as a promising paradigm for a large number of application domains such as environmental and medical monitoring [1] and home automation [2]. Many of these applications involve humans, where humans and things will operate synergistically [3]. At the heart of these systems is human monitoring where wearable sensors are utilized for sensing, processing, and transmission of physiological and contextual data. Typically, sensors acquire physical measurements, use computational models such as machine learning algorithms for local data processing, and communicate the results to a gateway or through the Internet [4].

Currently, an implicit assumption in development of machine learning algorithms for human-centered IoT applications is that the training and future data are in the same feature space and have the same distribution [5]. Therefore, most machine learning algorithms require significant amounts of training data for each application (e.g., activity recognition). Current research assumes that the machine learning algorithms need to be reconfigured (i.e., retrained) upon any changes in the configuration of the system, such as addition/ removal of a sensor to/from the network, displacement/ misplacement/ mis-orientation of the sensors, replacement/ upgrade of the sensors, adoption of the system by new users, and changes in physical/ behavioral status of the user. Retraining of the computational algorithms requires collecting sufficient amount of labeled training data, a time consuming, labor-intensive, and expensive process that has been identified as a major barrier to personalized and precision medicine [6, 7]. This problem becomes more challenging considering that wearables are deployed in highly dynamic and uncontrolled environments, due to their direct and continuous exposure to end-users and their living environments.

In this paper, we take first steps in developing automatic and real-time retraining of machine learning algorithms without labeled traning data. Specifically, we focus on cases where a new sensor in added to the system. Addressing the problem of expanding computational capabilities from single setting embedded software with predefined configuration to a dynamic setting where sensors can be added dynamically is challenging. In such cases, successful knowledge transfer is needed to improve the learning performance by avoiding expensive data collection, segmentation, and labeling efforts.

1.1 Related Work

Our work is inspired by transfer learning and specifically Teacher/ Learner instance transfer research[8]. TrAdaBoost [9], an extension of AdaBoost [10], proposed a boosting algorithm to enable transferring knowledge from one domain to another using training data from the source domain to incrementally build up the training data for the target domain. However, in the dynamic environment of human-centered IoT, the assumption that both source and target systems have the same feature space is unrealistic. Teacher/ Learner (TL) transfer learning has being used when there is no direct access to training data. Several studies [11, 12, 13] apply the teacher/ learner model to develop an opportunistic system capable of performing reliable activity recognition in a dynamic sensor environment. [13] showed that by synchronizing current sensor and new sensor, the existing sensor can provide labels of future activities. However, the accuracy of learner is bounded by the accuracy of teacher. In contrast, in this paper, we introduce a method for transferring and refining labels in order to outperform the accuracy of the initial system.

DAC '16, June 05-09, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2897937.2898066

1.2 Contributions

In this paper, we introduce the concept of *plug-n-learn* for human-centered IoT applications where machine learning algorithms are reconfigured automatically, in real-time, and without need for any new training data. Our pilot application in this paper is activity recognition where the goal is to detect physical movements of the user based on wearable sensor measurements. We address this problem by proposing a novel method to transfer activity recognition capabilities of one sensor to another where the collective network of the two sensors achieves much higher accuracy performance. Our approach allows to transfer machine learning knowledge from an existing sensor, called *source view*, to a new sensor, called *target view*, and these capabilities are augmented through the sensing observations made by the target view. We call this approach a Multi-view Learing and formulate this problem using Linear Programming, introduce a graph modeling of the problem, and propose a greedy heuristic to solve the problem. We evaluate the performance of the proposed automatic learning approach using real data collected with wearable motion sensors.

2. PROBLEM STATEMENT

Figure 1 shows the evolution of a wearable network as a new sensor is added to the system and a machine learning model (i.e., activity recognition classifier) is trained using unlabeled data captured by the new sensor. Initially, as shown in Figure 1a, the network consists of a fixed number of sensors (e.g., two sensors worn on 'Thigh' and 'Ankle') with trained machine learning models for activity recognition. We refer to the collection of the existing sensors as *'source view'*. A new sensor (e.g., a 'wrist-band') is added to the system as shown in Figure 1b. We refer to this newly added sensor as *'target view'*. Our multi-view learning approach captures sensor data in both views simultaneously, labels instances captured by the target based on the collective knowledge of source and target, and constructs a new classification algorithm for activity recognition. The system can now detect activities with higher accuracy in a collaborative fashion as shown in Figure 1c.

2.1 Problem Definition

An observation X_i made by a wearable sensor at time 'i' can be represented as an K-dimensional feature vector, $X_i = \{f_{i1}, f_{i2}, \ldots, f_{iK}\}$. Each feature is computed from a given time window and a marginal probability distribution over all possible feature values. The activity recognition task is composed of a label space $A = \{a_1, a_2, ..., a_m\}$ consisting of the set of labels for activities of interest, and a conditional probability distribution $P(A|X_i)$ which is the probability of assigning a label $a_j \in A$ given an observed instance X_i. Although a trained sensor can detect some activities, its observations are limited to the body segment on which the sensor is worn. Therefore, some activities are not recognizable in one sensor's view point. For example, a sensor worn on 'Ankle', although expert in detecting lower-body activities such as *walking*, cannot detect activities such as 'eating' that involves upper-body motions. To increase accuracy of activity recognition, it is desirable to add more sensors to the network to cover a larger set of physical activities. We note that advances in embedded sensor design and wearable electronics allow end-users to utilize new wearables such as smart watches and smart shoes. In such a realistic scenario, the newly added sensor has not been trained to collaborate with existing sensors in detecting physical activities. The general goal of our study is to develop an autonomous learning algorithm to detect activities that are not recognizable by existing sensors. Without loss of generality, we assume that the source view consists of a single sensor, in our formulation of this multi-view learning problem.

PROBLEM 1 (SYNCHRONOUS MULTI-VIEW LEARNING). *Let $S = \{s_1, \ldots, s_k\}$ be a set of k sensors. Let $s_r \in S$ be an existing sensor with a trained classification model. Furthermore, let $s_t \in S$ be a sensor newly added to the network. The sensors s_r and s_t are also referred to as source and target respectively. Moreover, let $A=\{a_1, a_2, \ldots, a_m\}$ be a set of m activities/labels that the system aims to recognize, and $X=\{X_1, X_2, \ldots, X_N\}$ a set of N observations made by s_t. The Synchronous Multi-View Learning (SML) problem is to accurately label observations made by s_t such that the mislabeling error is minimized. Once these instances are labeled, it is straightforward to develop a new activity recognition classifier using the labeled data.*

We note that transferring training data or classifier from 'source' to 'target' view is ineffective for the purpose of activity recognition. The main reason is that the feature spaces of the two sensors/views are completely different. Moreover, in practical applications, the training data is not available or the sensor is commercially trained and functions as a black-box and the training data are unavailable. Thus, our goal is to develop an approach that accurately labels observations made by the new sensor with assistance from the source sensor.

2.2 Problem Formulation

The SML problem described in Problem 1 can be formulated as follows.

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{j=1}^{m} x_{ij}\epsilon_{ij} \tag{1}$$

$$\text{Subject to:} \quad \sum_{i=1}^{N} x_{ij} \leq \Delta_i \ , \quad \forall j \in \{1, \ldots, m\} \tag{2}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \quad , \quad \forall i \in \{1, \ldots, N\} \tag{3}$$

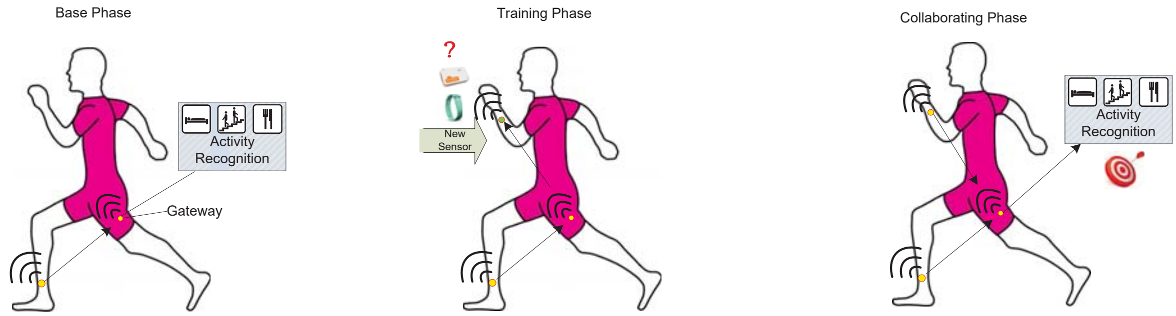
$$x_{ij} \in \{0, 1\} \quad , \quad \forall i, j \tag{4}$$

where x_{ij} is a binary variable indicating whether or not instance X_i is assigned activity label a_j, and ϵ_{ij} denotes error due to such a labeling. The constraint in (2) guarantees that a_j is assigned to at most Δ_i instances, where Δ_i is the upper bound on the number of instances with a_j as their possible labels. Since each instance must be assigned to exactly one actual activity label, the constraint in (3) guarantees that each instance X_i receives one and only one label. If ϵ_{ij} is known, this problem can be easily reduced to Generalized Assignment Problem. The main challenge in solving the SML problem is that the ground truth labels are not available in the target view and therefore ϵ_{ij} are unknown *a priori*.

Each sensor has a limited ability in detecting activities depending on the physical placement of the sensor on the body. That is, each sensor is expert in detecting some activities and is incapable of accurately detecting other activities. We define an 'ambiguity relation' to represent this concept.

DEFINITION 1 (AMBIGUITY RELATION). *An ambiguity relation AR is defined as a mapping $S \times A \rightarrow A$ such that for each sensor $s \in S$ and activity pair $a_1, a_2 \in A$, $((s, a_1), a_2) \in AR$ if and only if there exists a sensor s using which a_1 and a_2 are unrecognizable.*

We use the concept of ambiguity relation to determine deficiency of a sensor in detecting particular activities. We then resolve this recognition deficiency by combining information of multiple sensors through an exact labelling process in Section 3.



(a) Base setting with two sensors (b) Training newly added sensor (c) Collaboration

Figure 1 Evolution of a wearable network with plug-n-learn feature. Initially, the system consists of two sensors (a); Existing sensors, in the source view, contribute to training the new sensor, in the target view (b); When the training phase is completed, new sensor contributes to the system for Activity Recognition (AR), as shown in (c).

DEFINITION 2 (SEMI-LABEL). *A semi-label generated by sensor $s_r \in S$ is a 2-tuple (l_1, L) where $L \subset A$ is the set of all l_2 labels such that $((s, l_1), l_2) \in AR$. For example, if sensor s_1 is not able to reliably distinguish a_1 from a_2, it will generate a semi-label $(a_1, \{a_1, a_2\})$ to infer that the target activity can be either a_1 or a_2 rather than a_1. In this paper, we use confusion matrix of the classification algorithm used in the source view to retrieve semi-labels.*

2.3 Solution Overview

In our multi-view learning approach, the source sensor acts as a 'Teacher' and sends a semi-label of the current activity to the target sensor in real-time. The target sensor, also called 'Learner', learns from the information provided by 'Teacher' and boosts the labeling accuracy by combining its local observations with the received semi-labels. Thus, the challenging task is how to covert semi-labels to exact-labels such that the overall labeling error is minimized.

PROBLEM 2 (MINIMUM-ERROR EXACT LABELING). *Given a set of (instance, semi-label) pairs $\{(X_1,sl_1), (X_2,sl_2), \ldots, (X_n,sl_n)\}$ provided by a 'source' s_r, Minimum-Error Exact Labeling (MEEL) problem is to assign a label $a_j \in A$ to each instance X_i for all such instances such that the amount of mislabeling is minimized.*

Algorithm 1 Synchronous Multi-View Learning

---

while (more sensor data are required) **do**
  Source performs activity recognition on X_t at time t.
  Source detects activity a_t.
  Source computes *semi-label* sl_t for activity a_t.
  Source sends sl_t to the sensor in *Target* view.
  Target receives semi-label sl_t
  Target assigns sl_t to its observation at time t.
Use Minimum-Error Exact Labeling (MEEL) to compute an exact label for each semi-label

---

Algorithm 1 shows our multi-view learning approach for the optimization problem described in Problem 2. We assume that the sensors within the two views are worn on the body at the same time while the user performs physical activities. Because the source and target sensors may have different clocks, the target sensor needs to know to which observation each semi-label corresponds. We resolve the problem of different clocks by first synchronizing source and target sensors. At time t, the source sensor detects an instance of the current activity, a_t; it extracts corresponding semi-label sl_t and transmits (sl_t, t) to the target sensor via a wireless link. When the target sensor receives (sl_t, t), it assigns sl_t to its observation of time t. When sufficient instances of each activity are gathered, the MEEL problem is applied on the gathered data to label all instances in the target view.

3. MINIMUM-ERROR EXACT LABELING

We aim to use semi-labels provided by 'source' and develop a solution to the MEEL problem in order to enhance the labeling accuracy. To address this problem, we not only rely on the knowledge of 'source' (i.e., semi-labels) but also combine that knowledge with the observations made by 'target' to develop an accurate activity recognition model. Since the number of activities of interest is naturally less than the number of instances, we enhance the efficiency of our approach by breaking the solution into two steps: *instance partitioning* and *partition labeling*. In Figure 2, we provide an overview of our solution. Partitioning step is to find equivalence relation *Instance Equivalence Relation* on the set of instances such that each partition contains only instances of one activity. We assign an exact label to each partition with the goal of minimizing the amount of assignment error.

Assume a partitioning algorithm groups instances gathered by the target view into n partitions $C = \{C_1, C_2, \ldots, C_n\}$, $n \ll N$. We find optimal label for each cluster such that if the label of the partition propagates to the instances within that partition, the overall misclassification error is minimized. In this paper, we use a clustering algorithm as a heuristic approach to find *Instance Equivalence Relation*. The sensor in the 'target' view, based on its placement on the body, has some discriminating features to separate instances of ambiguous classes of the source sensor that are represented by semi-labels. Therefore, we use K-Means clustering [5] to partition instances of the 'target' view into n groups.

The linear programming formulation in Section 2.2 can be rewritten as follows.

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\epsilon_{ij} \tag{5}$$

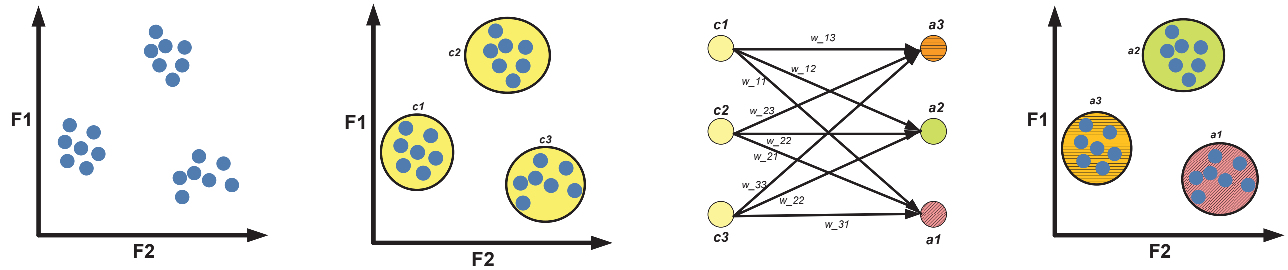
$$\text{Subject to:} \quad \sum_{i=1}^{n} y_{ij} = 1 \quad , \quad \forall j \in \{1, \ldots, m\} \tag{6}$$

$$\sum_{j=1}^{m} y_{ij} = 1 \quad , \quad \forall i \in \{1, \ldots, n\} \tag{7}$$

$$x_{ij} \in \{0, 1\} \quad , \quad \forall i, j \tag{8}$$

$$m = n \tag{9}$$

where y_{ij} is a binary variable indicating whether or not cluster C_i



(a) Target Instances Feature Space (b) Clustering Instances in Taget View (c) Weighted Labeling Graph (d) Label Propagation

Figure 2 Synchronous Multi-view Learning starts with (a) labeling target instances by semi-labels and continues with (b) partitioning target instances. Then, (c) a weighted bipartite graph is built on clusters and activities and after finding the best assignment, (d) cluster labels are propagated to their instances

is assigned label a_j, and ϵ_{ij} denotes the assignment error. With the clustering, we can estimate ϵ_{ij} by measuring the amount of disagreement in labels among individual instances within each clusters. Note that instances within each cluster already carry semi-labeling information provided by 'source'. Any conflict between the semi-label of an instance and its cluster label results in an assignment error for that instance. The constraint in (6) guarantees that a_j is assigned to exactly one cluster. It is reasonable to assume that, with sufficient accumulated data, the number of partitions is equal to the number of activities of interest. Therefore, we assume $n = m$ in the revised formulation. Each partition consists of *only* instances of one activity. Therefore, each partition can be assigned only one label. Additionally, each activity label can be assigned to one and only one partition, because partitions are expected to separate instances of different activities.

The aforementioned problem is in slake form and can be solved using *Simplex* algorithm whose time complexity is not polynomial [14]. However, constraints in (8) and (9) can transform the problem into a classic assignment problem. Because the nature of the problem is an assignment problem, it can be modeled using a bipartite graph and solved as a matching problem.

3.1 Weighted Labeling Graph

We construct a weighted complete bipartite graph to model the MEEL problem based on the clustered instances of the target view.

DEFINITION 3 (WEIGHTED LABELING GRAPH (WLG)). *Let C ={$C_1, \ldots, C_n$} be a set of clusters, each containing a set of instances for target sensor S_t. Furthermore, let A = {$a_1, \ldots, a_m$} be a set of given activity labels. A weighted labeling graph G(V,E,W) is a weighted complete bipartite graph such that V = {$C \cup A$} is the set of nodes in G such that $|C| = |A|$. E refers to the set of edges, and W is the set of edge weights such that an edge connecting cluster $C_i \in C$ to activity $A_j \in A$ has a weight of w_{ij} which is equivalent to the assignment error calculated by measuring the amount of disagreement between each instance's semi-label and the label of the cluster that instance belongs to. For impossible assignments $w_{ij} = \infty$.*

In a simple example shown in Figure 2, the semi-labels provided by 'source' are used to label the 'target' observations shown in Figure 2a. First, the data instances are clustered as shown in Figure 2b. The clustering results are then used to construct a labeling graph shown in Figure 2c.

3.2 Labeling Algorithm

We propose a simple polynomial time algorithm to solve the MEEL problem. Our goal is to assign exactly one class label to each partition such that the total cost of the assignment is minimized. This problem can be viewed as a weighted matching problem. Since $G(V,E,W)$ is a complete bipratite graph and $|C| = |A|$, we find a perfect matching in G to minimize the assignment costs (Figure 2c). Although there are different methods to solve this assignment problem, our approach is based on the well-known *Hungarian* algorithm [15] which runs in $O(n^3)$ where $n = |C| = |A|$. Finally, we propagate assigned label of the partitions to their instances (Figure 2d).

Algorithm 2 Greedy Algorithm for MEEL

INPUT: Set (X, SL)
OUTPUT: finalSet
PA ← getPossibleLabels(SL)
C ← cluster(X)
G ← buildBipartiteGraph(C, PA)
(C, A) ← findBestAssignment(G)
finalSet ← { }
for each (Ci, Aj) in (C, A) **do**
  for each Xk in Ci **do**
    add (Xk, Aj) → finalSet

Input to our labeling algorithm, shown in Algorithm 2, is (X, SL), a set of all target instances and their corresponding semi-labels. Given that each semi-label associated with an instance is a set of potential labels for that instance, we define PA as the union of all such label sets. The algorithm proceeds by clustering all instances based on the feature space of the target sensor. Next, it constructs a weighted labeling graph, as discussed in Definition 3, from all clusters and all labels. In the next step, $findBestAssignment$ executes the *Hungarian* algorithm on the constructed graph and generates a set of assignments, one for each cluster. Finally, we propagate the cluster labels to their corresponding instances and generate the final set (X, TL) of instances and their transferred labels.

4. EXPERIMENTAL RESULT

In this section, we demonstrate the efficiency of our multi-view learning algorithm using real data collected from 3 human subjects performing 14 different physical activities while wearing 3 wireless motion sensor nodes on their 'Left Ankle' (LA), 'Right Wrist' (RW) and 'Waist' (Wa). Each sensor node had a 3-axis accelerometer and 2-axis gyroscope. The physical activities included *Stand to Sit* (St2S), *Sit to Stand* (S2St), *Sit to Lie* (S2L), *Lie to Sit* (L2S), *Jump* (Jmp), *Turn Clockwise* (TC), *Bend to Grasp* (B2G), *Rising from Bending* (RfB), *Step Forward* (SF), *Step Backward* (SB), *Look Back* (LB), *Kneeling Right* (KR), *Rise from Kneeling* (RfK) and *Return from Looking back* (RfL). The sampling frequency was set

Table 1 Accuracy (%) of single-node activity recognition

Classifier	LA	RW	Wa
J48	44.28	52.14	48.57
SVM	55.01	37.85	67.85
kNN	**57.85**	**69.28**	**73.57**

to 50Hz and the subjects were asked to repeat each activity 10 times while the data were being collected wirelessly. The collected dataset contained over 630, 000 samples of acceleration and angular velocity. The data were used to extract 10 statistical features from the individual sensor streams.

4.1 Single-Node Performance

Our initial analysis focused on assessing performance of the system in a single-node (i.e., 'Left Ankle', 'Right Wrist' or 'Waist') scenario. We used each one of these sensors to detect all activities using three machine learning classifiers, namely Decision Tree (J48), Support Vector Machines (SVM), and k-Neareset-Neighbor (kNN). Table 1 shows the activity recognition accuracy for this analysis. The average accuracy of 'Left Ankle', 'Right Wrist', and 'Waist' sensors was 52.38%, 53.09%, and 63.33% respectively. These results suggest that we cannot achieve a high recognition accuracy using only one sensor. This finding confirms our hypothesis that, in a single sensor view, some activities are not reliably distinguishable. Furthermore, the ability of different sensors in detecting different activities varies. For example, as shown in Table 1, 'Left Ankle' is weaker than 'Waist' in detecting all 14 activities regardless of the selected classification algorithm.

4.2 Comparative Evaluation Method

Research in the area of transfer learning for wearables is new. To the best of our knowledge, there exist only two of such algorithms, namely Naive and System-Supervised, suggested in [13], which are applicable to the synchronous teacher/ learner approach studied in this paper. We compare our multi-view learning technique with these two algorithms as well as the experimental upper bound obtained using ground truth labels gathered during our data collection. Calatroni et. al [13] proposed the Naive approach as reusing the 'source' classifier in 'target'. They emphasized that this method only works when 'source' and 'target' are completely similar (e.g., sensors are co-located on the body and are homogeneous). The *System-Supervised* method refers to the case where 'target' labels its observations based on labels predicted by 'source'. Our analysis compares the accuracy of provided labels in 'target' using SML with that of Naive, System-Supervised, and Upper-Bound. We also report the accuracy of the target classifier built on the computed labels using all the algorithms.

Table 2 Semi-labels and unrecognizable activities

Sensor Node	Semi-label	
	Predicted Class	Possible Actual Classes
Left Ankle (LA)	St2S	St2S, L2S
	S2St	St2S, S2St, TC
	L2S	S2L
	TC	L2S
	KR	KR, RfK
Right Wrist (RW)	LB	SF, LB
	KR	RfB, RfK
	RfK	B2S, KR, RfK
Waist (Wa)	B2S	S2St, B2S
	RfB	St2S, RfB
	SF	SF, LB
	KR	KR, RfK

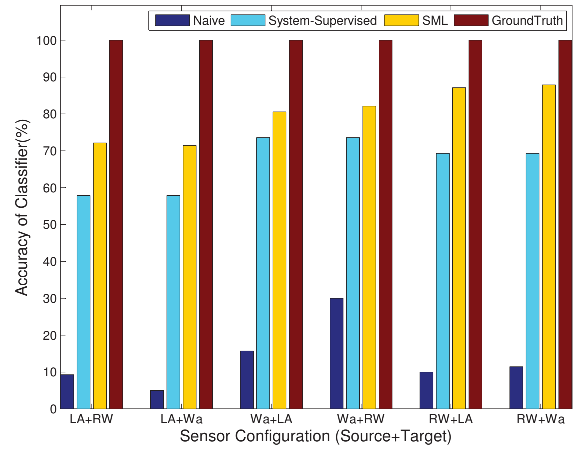


Figure 3 Accuracy of labeling for four approaches under comparison

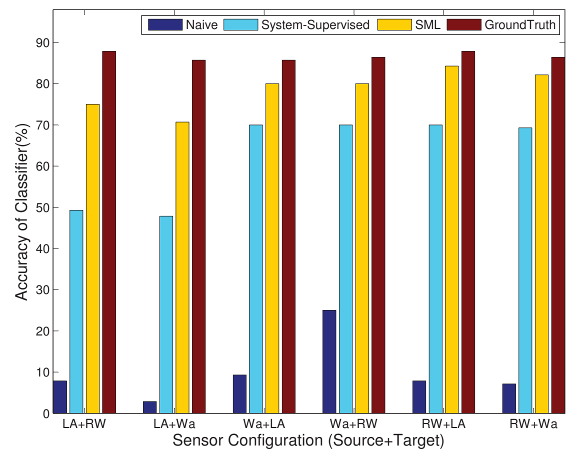
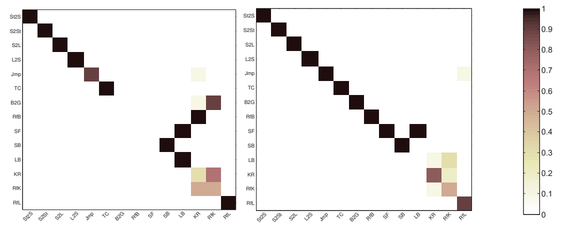


Figure 4 Accuracy of four activity recognition classifiers

4.3 Comparative Analysis

We studied different scenarios where 'source' and 'target' could be any of the three sensors, namely 'Left Ankle' (LA), 'Right Wrist' (RW) or 'Waist' (Wa). For each scenario, we utilized the confusion matrix of the source classifier to extract semi-labels. For example, when 'Left Ankle' detects an unknown instance as *Kneeling Right (KR)*, the actual activity could be one of *Kneeling Right (KR)* or *Rising from Kneeling (RfK)* as shown in Table 2. Furthermore, by studying the confusion matrix of different classifiers, we noticed that failing to detect an activity is more related to the position of the sensor rather than the type of algorithm or classifier used for activity recognition. Once semi-labels are extracted, we followed the rest of our SML procedure described in Algorithm 1 to transfer semi-labels and to compute exact labels at the target view. The algorithm clusters target instances based on the statistical features extracted from signals in the target view and continues with applying MEEL to find the optimal labeling.

As shown in Figure 3, the accuracy of labeling using SML in all six scenarios is higher than 'Naive' and 'System-Supervised' methods. The overall accuracy of labeling using SML ranged from 71.42% to 87.86% with an average accuracy of 80.21%. This accuracy is 65.6% and 13.3% higher than the accuracy of 'Naive' and 'System-Supervised' respectively. We note that the accuracy of the



(a) Source Sensor (b) After SML (c) scale

Figure 5 Comparison of confusion matrices before and after applying SML, when 'source' is 'Right Wrist (RW)' and 'target' is 'Waist (Wa)'

ground truth labeling is naturally 100%. This ground truth refers to labeling of the data by examining the activity types using video recordings of the experiments.

After labels are obtained in the target view, we build a kNN classifier on those labels for activity recognition. As illustrated in Figure 4, the classification accuracy of our SML approach ranges from 70% to 82.15%. Compared to 'system-supervised', SML improved the recognition accuracy by 15.96%, on average, without requiring any labeled training data. Furthermore, SML performs 68.69% better than 'Naive' method in terms of activity recognition accuracy. In addition, we computed the experimental upper bound of the two-sensor configuration where we used ground truth labels to train a classifier using both source and target sensors. The overall experimental upper bound was 86.66% averaged over all experimental configurations. Therefore, the recognition accuracy of our approach is only 7.96% lower than the experimental upper bound.

The highest improvement in labeling is achieved for the 'RW+Wa' scenario where 'Right Wrist' operates as 'source' and 'Waist' as 'target'. Figure 5 demonstrates how SML improves the provided labels in the target by comparing confusion matrix of provided labels using 'source' (system-supervised) and the confusion matrix of labels after applying SML. As shown in Figure 5b, after applying SML, more instances are concentrated on the main diagonal of the confusion matrix, as specified by darker cells in the graph.

5. DISCUSSION AND FUTURE WORK

Our study in this paper is a first step towards designing next generation wearables that are computationally autonomous and can automatically learn machine learning algorithms. Dynamic attributes of wearables are not limited to real-time sensor addition. A sensor can be be removed, misplaced, displaced, upgraded, or replaced, sensors can produce noisy signals, and the system can be adopted by new users. Our ongoing research involves development of transfer learning algorithms that address dynamically evolving behavior of human-centered monitoring applications.

In this paper, we evaluated SML on sensor addition scenarios. We believe that our SML technique is applicable to any scenarios where 'source' and 'target' observe the phenomena of interest synchronously. We plan to investigate scenarios such as removal or on-body relocation of the sensors in real-time, or replacement with asynchronous sensors. We also plan to study the the effectiveness of our multi-view learning approach in a network with heterogeneous sensors.

Our results show that the accuracy of a classifier using one sensor such as 'Left Ankle' was extremely lower than the accuracy of the classifier trained using 'Right Wrist' or 'Waist'. After applying our multi-view learning, however, the performance measures are comparable. This in fact indicates that our approach works more efficient when the source sensor is less accurate because it can cover blind spots of the source sensors.

6. CONCLUSION

As wearable sensors are becoming more prevalent, their function becomes more complex and they operate in highly dynamic environments. Machine learning algorithms for these sensors cannot be designed only for one specific setting. To address the dynamic nature of wearable sensors, we proposed a multi-view learning approach that uses the knowledge of existing sensors to adapt with sensor addition. We used activity recognition task as our pilot application. We introduced a multi-view earning approach to learn computational algorithms in new settings without any need for labeled training data in the target view. Our experiments show that we can combine knowledge of existing system with the observation of a new sensor to boost the accuracy of the system without labeled training data with the new sensor. Our results demonstrate promising results with an overall accuracy, which is only 7.96% lower than practical upper bound.

7. REFERENCES

[1] U. Anliker, J. Ward, Lukowicz *et al.*, "Amon: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 4, pp. 415–427, Dec 2004.

[2] K. D. Feuz and D. J. Cook, "Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr)," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 1, p. 3, 2015.

[3] J. Stankovic, "Research directions for the internet of things," *Internet of Things Journal, IEEE*, vol. 1, no. 1, pp. 3–9, Feb 2014.

[4] H. Ghasemzadeh, N. Amini, R. Saeedi, and M. Sarrafzadeh, "Power-aware computing in wearable sensor networks: An optimal feature selection," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 4, pp. 800–812, April 2015.

[5] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[6] D. Zakim and M. Schwab, "Data collection as a barrier to personalized medicine," *Trends in pharmacological sciences*, vol. 36, no. 2, pp. 68–71, 2015.

[7] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing*. Springer, 2004, pp. 1–17.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[9] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 193–200.

[10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[11] D. Roggen, K. FÃűrster, A. Calatroni, and G. TrÃűster, "The adarc pattern analysis architecture for adaptive human activity recognition systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 169–186, 2013.

[12] M. Kurz, G. Holzl, A. Ferscha, A. Calatroni, D. Roggen, G. Troster, Sagha *et al.*, "The opportunity framework and data processing ecosystem for opportunistic activity and context recognition," *International Journal of Sensors Wireless Communications and Control*, vol. 1, no. 2, pp. 102–125, 2011.

[13] A. Calatroni, D. Roggen, and G. Tröster, "Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion," *Eighth international conference on networked sensing systems (INSS'11), Penghu, Taiwan*, vol. 6, 2011.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.

[15] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171 – 175, 1986.