

Instance-Wise Dynamic Sensor Selection for Human Activity Recognition

Xiaodong Yang,^{1,2,3} Yiqiang Chen,^{1,2,3*} Hanchao Yu,^{1*} Yingwei Zhang,^{1,2}
Wang Lu,^{1,2,3} Ruizhe Sun^{1,2}

¹The Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, 100190

²University of Chinese Academy of Sciences, Beijing, China, 100049

³Peng Cheng Laboratory, Shenzhen, China, 518055

{yangxiaodong, yqchen, yuhanchao, zhangyingwei, luwang, sunruizhe18s}@ict.ac.cn

Abstract

Human Activity Recognition (HAR) is an important application of smart wearable/mobile systems for many human-centric problems such as healthcare. The multi-sensor synchronous measurement has shown better performance for HAR than a single sensor. However, the multi-sensor setting increases the costs of data transmission, computation and energy. Therefore, the efficient sensor selection to balance recognition accuracy and sensor cost is the critical challenge. In this paper, we propose an *Instance-wise Dynamic Sensor Selection (IDSS)* method for HAR. Firstly, we formalize this problem as minimizing both activity classification loss and sensor number by dynamically selecting a sparse subset for each instance. Then, IDSS solves the above minimization problem via Markov Decision Process whose policy for sensor selection is learned by exploiting the instance-wise states using Imitation Learning. In order to optimize the parameters of the activity classification model and the sensor selection policy, an algorithm named *Mutual Dagger* is proposed to alternatively enhance their learning process. To evaluate the performance of IDSS, we conduct experiments on three real-world HAR datasets. The experimental results show that IDSS can effectively reduce the overall sensor number without losing accuracy and outperforms the state-of-the-art methods regarding the combined measurement of accuracy and sensor number.

Introduction

Human Activity Recognition (HAR) aims to recognize an individual's daily activities, such as standing, walking and cycling. There is a considerable demand for applying HAR on wearable devices to understand human behavior and provide services like activity tracker (Tang et al. 2018), healthcare monitoring (Chen et al. 2017; Yu et al. 2015), assisted rehabilitation (Mazilu et al. 2013), et al. To deal with the complexity and variety of daily activities, HAR in real-world environments requires multi-sensor synchronous measurement, e.g., Body Sensor Networks (BSN), which contains several sensors with different modalities and being worn on different body parts. However, more sensors bring more cost

of massive data transmission, cumbersome computational load, and large energy consumption, which limits the operational lifetime span (Kolamunna et al. 2016). Therefore, to balance recognition accuracy and sensor number becomes an emergent challenge for HAR.

Most existing HAR methods use a fixed set of sensors for data acquisition, which either collects redundant sensors for highly discriminable instances or obtains unsatisfactory accuracy for confusing instances. There are a few works on seeking a dynamic sensor selection method. Gordon et al. (Gordon et al. 2012) propose a method which weights the sensors to the target activities, and then decides which sensors should be activated based on the recent prediction. Some works in camera networks (Spaan and Lima 2009; Satsangi, Whiteson, and Oliehoek 2015) solve the dynamic sensor selection problem with a discrete state Partially Observable Markov Decision Process (POMDP). They merely take one of the predefined states as input and return which sensors should be active. These methods lack direct information from the instance, and cannot make the desired performance for balancing recognition accuracy and sensor number. Since sensor selection can be categorized into feature selection, some methods (He, Daumé III, and Eisner 2012; Karayev, Fritz, and Darrell 2013; Kolamunna et al. 2016) attempt to build an MDP to decide which feature to select next or whether to stop acquiring features and make a prediction. Although selecting features dynamically within a timestep, these methods condition the choice of one feature on the observation generated by another one. This setting is not feasible in the real-time HAR where all selected sensors must be chosen simultaneously.

In this paper, we propose an effective method named Instance-dependent Dynamic Sensor Selection (IDSS) for HAR. Specifically, given a sequence of activity instances, our goal is to select an appropriate subset of sensors for each instance to predict the activity, so that recognition accuracy and sensor number are balanced in a desired way. We formalize the problem as seeking an activity classification model and a sensor selection policy that minimizes an objective function of overall classification loss and sensor number. IDSS defines a continuous state MDP to address the minimization problem, where the policy function for sensor

*Corresponding author

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

selection is learned using Imitation Learning by building a direct mapping between the sensors and the instance-wise states. To optimize the parameters of the activity classification model and the sensor selection policy, we propose an algorithm named Mutual DAgger, where the two parts of parameters are alternatively trained with mutually augmented data. Therefore, our main contributions can be summarized as follows:

- We formalize a problem which finds an activity classification model and a sensor selection policy to minimize an objective function regarding classification loss and sensor number.
- We propose IDSS to solve the minimization problem. IDSS utilizes a continuous state MDP where Imitation Learning is employed to model the policy function to determine each sensor whether to activate by directly exploiting the information of the latest instance.
- IDSS unifies the parameters of the classification model and the selection policy in one model and optimizes them by the proposed Mutual DAgger which builds an alternative learning process with mutually augmented data.

To evaluate the performance of IDSS, we conduct extensive experiments on three real-world HAR datasets. The results demonstrate that IDSS outperforms the state-of-the-art methods regarding a combined measurement of recognition accuracy and sensor number.

Related Works

Human Activity Recognition has been widely investigated as an important topic in artificial intelligence. The development of wearable technology and pervasive computing prompts a substantial number of applications based on HAR by enabling low cost, continuous and non-invasive mobile sensing (Gravina et al. 2017). Most methods for HAR use all the available sensors to gain a high recognition accuracy, but a large number of sensors bring much resource consumption in bandwidth, storage and power. As resources are limited in mobile environments, how to balance the recognition accuracy and resource consumption is one of the significant problems for HAR.

Sensor selection, which can be categorized into feature selection, is one of the effective ways to reduce consumption without losing performance. Shen et al. (Shen and Varshney 2013) used a generalized Information Gain to select a fixed set of sensors for target tracking. For recognizing the human activity, some works (Aziz, Robinovitch, and Park 2016; Ertuğrul and Kaya 2017; Cheng et al. 2018) identified the optimal number and position of sensors by determining the relevancy ranks of the sensors through filtering methodology. However, the goal of the fixed sensor selection is to optimize the overall accuracy, which results in either collecting redundant information for highly discriminable instances or obtaining unsatisfactory accuracy for confusing ones. Therefore, selecting sensors/features dynamically for each instance has been proposed to gain a better trade-off between accuracy and sensor number. Bloom et al. (Bloom, Argyriou, and Makris 2013) pre-defined some feature subsets and selected the best prediction to be the output. Zappi

et al. (Zappi et al. 2008) proposed a method to build a meta-classifier for each sensor and select sensors according to their contribution to classification accuracy. Gordon et al. (Gordon et al. 2012) weighted the importance of each sensor to the activities and only sampled those sensors which were related to the potential activities. As dynamic sensor selection in continuous HAR is a sequential decision process, these methods use the immediate rewards to make the decision cannot gain the desirable solution on the activity sequence.

Many works regard this issue as a Markov Decision Process. Some of them (Dulac-Arnold et al. 2011; He, Daumé III, and Eisner 2012; Karayev, Fritz, and Darrell 2013; Trapeznikov and Saligrama 2013) conditioned the choice of one sensor/feature on the observation generation by another sensor/feature within a timestep. In each timestep, they assume that all the sensors/features are available and the MDP decides which sensor/feature to consider or whether to stop acquiring and make the inference. However, in the real world, the observations of sensors are generated in parallel and all the selected sensors must be chosen simultaneously. In this setting, other works build MDP along with the timesteps instead of within a timestep. For target tracking in camera networks, Spaan et al. (Spaan and Lima 2009) and Satangi et al. (Satsangi, Whiteson, and Oliehoek 2015) proposed to select camera sensors through Partially Observable Markov Decision Processes which discretize the continuous state (the position of target) to model the policy. However, the state discretization is not feasible for wearable sensors, and they do not take advantage of the direct information of sensors.

Besides sensor selection, there are also some other techniques for balancing the accuracy and sensor cost, e.g., adaptive sampling frequency (Cheng et al. 2018) and compressive sensing (Wang et al. 2016), which are orthogonal to our work. In this paper, we focus on reducing the number of required sensors, which can work with the aforementioned techniques.

Methodology

In this section, we first formalize the problem of balancing activity recognition accuracy and sensor cost as minimizing an objective function regarding classification loss and sensor number. Then, we propose an Instance-wise Dynamic Sensor Selection (IDSS) method for HAR, which recognizes the activity based on the dynamic sensors selected by a continuous state MDP.

Problem Formalization

Given K sensors in the system, there are $(2^K - 1)$ available subsets of sensors to sample an instance for recognizing the ongoing activity. Let $\mathbf{z} = \langle z_1, \dots, z_K \rangle$ be a vector of K binary elements, each of which specifies whether the corresponding sensor is selected or not. Let \mathbf{x} be an instance which contains a series of samples from all the sensors, and $y \in \{1, 2, \dots, m\}$ be the activity label where m is the number of predefined activities. We define a sampling function $g(\mathbf{x}, \mathbf{z})$ which returns an observed instance $\hat{\mathbf{x}} = g(\mathbf{x}, \mathbf{z})$ by

sampling \mathbf{x} with the selected sensors. Given an $\tilde{\mathbf{x}}$, we can predict its label by a classification model parameterized by θ and get the probabilities $\mathbf{p} \in [0, 1]^m$ for m categories:

$$\mathbf{p} = \langle p_1, \dots, p_m \rangle \quad (1)$$

where the y -th element p_y is the probability of the activity y :

$$p_y = p(y|\tilde{\mathbf{x}}; \theta) \quad (2)$$

Let $Q = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(T)}, y^{(T)})\}$ be a sequence of annotated activity instances, where T is the horizon. On one hand, our goal is *accurate activity recognition* which means to optimize θ of the classification model by minimizing the empirical loss; on the other hand, our goal is to recognize activities with *a minimal number of sensors* for reducing sensor cost.

We consider simultaneously the classification loss $\mathcal{L}(\hat{y}^{(t)}; \tilde{\mathbf{x}}^{(t)}, \theta)$ and the sensor number $\|\mathbf{z}^{(t)}\|_1$ with a balance parameter λ to obtain an overall objective function:

$$\min_{\theta, \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(T)}} \sum_{t=1}^T \mathcal{L}(\hat{y}^{(t)}; \tilde{\mathbf{x}}^{(t)}, \theta) + \lambda \frac{\|\mathbf{z}^{(t)}\|_1}{K} \quad (3)$$

where $\hat{y}^{(t)} = \arg \max_y p(y|\tilde{\mathbf{x}}^{(t)}, \theta)$ and $\mathcal{L}(\cdot)$ is a loss function, such as the cross-entropy in Neural Networks or the hinge loss in SVM. The measurement of sensor number is normalized into $(0, 1]$. λ is limited to $[0, 1]$ for the dominant effect of activity recognition accuracy. When $\lambda = 0$, the problem is a conventional supervised learning issue; when $\lambda = 1$, the sensor cost is emphasized as equally as the recognition accuracy. In this work, cross-entropy is used for presenting the classification loss, then Equation 3 can be further rewritten as:

$$\min_{\theta, \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(T)}} \sum_{t=1}^T -\log p(y^{(t)}|\tilde{\mathbf{x}}^{(t)}; \theta) + \lambda \frac{\|\mathbf{z}^{(t)}\|_1}{K} \quad (4)$$

To optimize the defined objective function, there are two parts of parameters to be learned, one of which is for activity recognition and the other is for sensor selection.

Instance-wise Dynamic Sensor Selection (IDSS)

IDSS aims to select an appropriate sensor subset for each incoming instance to make an accurate activity recognition. As aforementioned, dynamic sensor selection is a sequence of decisions. Inspired by the existing works, IDSS solves the minimization problem of Equation 4 by employing a Markov Decision Process. Considering that the diversity exists not only among activities but also among instances within one activity, it is essential to learn a policy π which builds a direct mapping from the instance-wise states to the sensor-selection actions. In this section, we define a continuous state MDP and prove its feasibility to solve the formalized problem. Then, we employ Imitation Learning to learn the policy function of MDP and propose an alternative enhancement algorithm *Mutual DAgger* to optimize the parameters of the activity classification model as well as the sensor selection policy.

The Markov Decision Process in IDSS

Let $\phi(\tilde{\mathbf{x}}) \in \mathbb{R}^d$ denotes the feature vector extracted from $\tilde{\mathbf{x}} = g(\mathbf{x}, \mathbf{z})$ where d is the dimension. Since dynamically selected sensors lead to an issue of various feature dimensions, we use a unified length of the feature vector, and the unobserved features are imputed by a constant value, e.g., 0. We introduce the MDP as a tuple $\langle \mathcal{S}, \mathcal{A}, T(\cdot), R(\cdot) \rangle$:

- \mathcal{S} is an infinite state space. Each state $\mathbf{s} \in \mathcal{S}$ is defined as a vector $\mathbf{s} = (\phi(\tilde{\mathbf{x}}), \mathbf{p}) \in \mathbb{R}^{d+m}$, which is concatenation of the extracted features and the activity probabilities. Under this definition, $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \mathbf{p}^{(t)})$ describes the system state at the t timestep.
- \mathcal{A} is a set of actions, each of which indicates which sensors are selected. For brevity, we directly define $\mathcal{A} := \{\mathbf{z}\}_1^{2^K-1}$.
- $T(\mathbf{s}'|\mathbf{s}, \mathbf{z})$ is the transition probability function. Given a state $\mathbf{s} \in \mathcal{S}$ and an action $\mathbf{z} \in \mathcal{A}$, $T(\mathbf{s}'|\mathbf{s}, \mathbf{z})$ indicates the probability of the next state reached being \mathbf{s}' by taking the action \mathbf{z} . For the given instance sequence for training, the transition of the states is defined as:

$$T(\mathbf{s}'|\mathbf{s}, \mathbf{z}) = \begin{cases} 1, & \mathbf{s}' = \mathbf{s}^{(t)}, \mathbf{s} = \mathbf{s}^{(t-1)}, \mathbf{z} = \mathbf{z}^{(t-1)} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

- $R(\mathbf{s})$ is the reward function. It represents a reward by visiting a state $\mathbf{s} = (\phi(\tilde{\mathbf{x}}), \mathbf{p})$. Specifically, the reward function can be defined as:

$$R(\mathbf{s}) = \log p(y|\tilde{\mathbf{x}}; \theta) - \lambda \frac{\|\mathbf{z}\|_1}{K} \quad (6)$$

where the selected sensors are implicitly contained in the state. We define $R(\mathbf{0}) = 0$ where $\mathbf{0}$ is a zero vector.

The dynamics of the MDP can be described as follows: the system is started at an initial state $\mathbf{s}^{(0)} = \mathbf{0}$, and a first action $\mathbf{z}^{(0)}$ is chosen to decide which sensors are going to activate and the others are deactivated. Then, the corresponding sensors are sampled as the first instance $\tilde{\mathbf{x}}^{(1)}$. With the classification model to fetch $\mathbf{p}^{(1)}$, the first state is stepped into $\mathbf{s}^{(1)} = (\phi(\tilde{\mathbf{x}}^{(1)}), \mathbf{p}^{(1)})$. After that, the system chooses a second action $\mathbf{z}^{(1)}$ based on $\mathbf{s}^{(1)}$ by:

$$\mathbf{z}^{(t)} = \arg \max_{\mathbf{z} \in \mathcal{A}} V(\mathbf{s}^{(t)}) \quad (7)$$

where V is the value function of states, which is defined in a recursive way:

$$V(\mathbf{s}^{(t)}) = R(\mathbf{s}^{(t)}) + \int_{\mathbf{s}^{(t+1)}} T(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, \mathbf{z}) V(\mathbf{s}^{(t+1)}) \quad (8)$$

When $t = T$, $V(\mathbf{s}^{(T)}) = R(\mathbf{s}^{(T)})$. By taking the new action, the system goes forward to the next state. The procedure keeps repeating until the horizon. Finally, a trajectory denoted by τ of state/action/reward is formed:

$$\tau = \mathbf{s}^{(0)}, \mathbf{z}^{(0)}, r^{(0)}, \mathbf{s}^{(1)}, \mathbf{z}^{(1)}, r^{(1)}, \dots, \mathbf{s}^{(T)}, \mathbf{z}^{(T)}, r^{(T)} \quad (9)$$

where $r^{(t)} = R(s^{(t)})$. Thus, the total reward of the trajectory is consequently defined as:

$$R(\tau) = \sum_{t=1}^T r^{(t)} = R(\mathbf{0}) + \sum_{t=1}^T R(s^{(t)}) \quad (10)$$

The ultimate objective function of MDP is to make the right decision on each action to build a trajectory with the maximal total reward. Therefore, the objective function of MDP can be written as:

$$\begin{aligned} & \max_{z^{(1)}, \dots, z^{(T)}} R(\tau) \\ = & \max_{z^{(1)}, \dots, z^{(T)}} R(\mathbf{0}) + \sum_{t=1}^T R(s^{(t)}) \\ = & \max_{z^{(1)}, \dots, z^{(T)}} \sum_{t=1}^T R(s^{(t)}) \\ = & \max_{z^{(1)}, \dots, z^{(T)}} \sum_{t=1}^T \log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \boldsymbol{\theta}) - \lambda \frac{\|z^{(t)}\|_1}{K} \\ \iff & \min_{z^{(1)}, \dots, z^{(T)}} \sum_{t=1}^T -\log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \boldsymbol{\theta}) + \lambda \frac{\|z^{(t)}\|_1}{K} \end{aligned}$$

By the deduction process, the goal of MDP that maximizing the total reward is equivalent to the original minimization problem shown in Equation 4. In other words, MDP can be used to solve the problem of balancing the recognition accuracy and the sensor cost.

Modeling the policy in MDP by Imitation Learning

To model its policy function π which decides the action based on the state, we find the optimal value of each state by Bellman Equation (Bellman 1952) in a dynamic programming manner:

$$\begin{aligned} V^*(s^{(t)}) = & R(s^{(t)}) + \\ & \max_{z \in \mathcal{A}} \int_{s^{(t+1)}} T(s^{(t+1)} | s^{(t)}, a) V^*(s^{(t+1)}) \end{aligned} \quad (11)$$

Corresponding to V^* , the optimal policy function π^* regarding s is calculated as:

$$\pi^*(s) = \arg \max_{z \in \mathcal{A}} \int_{s'} T(s' | s, z) V^*(s') \quad (12)$$

Assuming we have a fixed classification model, we can get a series of state-action pairs $\mathcal{H} = \{(s, z)\}$ by searching through the activity sequence. The searching process is a cumbersome task since the MDP is an NP-hard problem, and the space of action increases exponentially with the number of available sensors. In this paper, we employ a greedy maximization process whose error upper bound is proven by Satsangi et al. (Satsangi, Whiteson, and Oliehoek 2015).

Since the instance-wise states of MDP are continuous and infinite, the conventional tabular-based and state discretizing methods are not applicable for modeling the policy. What's

more, the transition of states is hard to model for those unseen activity sequences. As the obtained state-action pairs are hindsight experienced, they can be regarded as demonstrations observed by an expert. Thus, Imitation Learning by classification, such as Behavioral Cloning (Osa et al. 2018), is employed to learn a policy function π parameterized by ψ through minimizing the imitating loss:

$$\min_{\psi} \mathbb{E}_{(s, z) \sim \mathcal{H}} [-\log p(z | s; \psi)] \quad (13)$$

Instead of building a $(2^K - 1)$ -class classification, we design a K -output classifier by combining the binary classification loss of each sensor together:

$$\min_{\psi} \mathbb{E}_{(s, z) \sim \mathcal{H}} \left[\sum_{k=1}^K -\log p(z_k | s; \psi) \right] \quad (14)$$

which avoids some issues such as there may be some actions never be chosen in \mathcal{H} .

Mutual DAgger for parameter optimization

There are two parts of parameters to optimize in the proposed IDSS: the classification model parameter $\boldsymbol{\theta}$ and the MDP policy parameter ψ , which are influence by each other. The conventional supervised learning is not desirable in the dynamic sensor setting: on one hand, the classification model $\boldsymbol{\theta}$ should be fed various data sampled from different sensor subsets; on the other hand, the selection policy ψ should be able to recover from an error selection.

Inspired by the existing works, such as DAgger (Ross, Gordon, and Bagnell 2011) which designs a data augmentation process to enable the learned model recover from errors. We propose an algorithm named *Mutual DAgger* which builds a mutual data augmentation during the alternative learning between the classification model and the selection policy. Given ψ and a sequence of activity instances, a new series of actions are made according to the action policy of the defined MDP. Then, the entities with new features $Q' = (\tilde{\mathbf{x}}', y)$ can be fetched to augment the training dataset \mathcal{Q} for $\boldsymbol{\theta}$ by minimizing the surrogate loss:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\tilde{\mathbf{x}}, y) \sim \mathcal{Q} \cup Q'} [-\log p(y | \tilde{\mathbf{x}}; \boldsymbol{\theta})] \quad (15)$$

In this way, $\boldsymbol{\theta}$ is fine-tuned based on ψ . Alternatively, given $\boldsymbol{\theta}$, more state-action pairs \mathcal{H} are extracted based on Equation 12. The new pairs provide new situations of states which includes the potential error selections. By augmenting its training dataset, ψ can be further fine-tuned by :

$$\min_{\psi} \mathbb{E}_{(s, z) \sim \mathcal{H} \cup \mathcal{H}'} \left[\sum_{k=1}^K -\log p(z_k | s; \psi) \right] \quad (16)$$

In *MaxIter* rounds, $\boldsymbol{\theta}$ and ψ are optimized by the alternative learning process. Up to now, the pseudo-code is concluded in Algorithm 1.

IDSS Inference

After learning the parameters of the classification model $\boldsymbol{\theta}^*$ and the selection policy ψ^* , we can use IDSS to select a sensor subset for the incoming instance and infer its activity. We

Algorithm 1 IDSS Learning with Mutual DAGger

Input:

A instance sequence with labels $Q = \{(\mathbf{x}, y)\}$;
The iteration $MaxIter$;

Output:

The classification model parameter θ ;
The selection model parameter ψ ;

```

1: Initialize  $z_{all} = \{1\}^K$ ,  $\mathcal{H} = \Phi$ ,  $\mathcal{Q} = \{(\mathbf{x}, y, z_{all})\}$ ;
2: Learning  $\theta$  by Equation 15 using  $\mathcal{Q}$ ;
3: while  $iter = 1, 2, \dots, MaxIter$  do
4:   Fix  $\theta$ , obtain the state-action pair  $H' = \{(s, a)\}$  by
      Equation 12;
5:    $\mathcal{H} = \mathcal{H} \cup H'$ ;
6:   Update  $\psi$  by Equation 16 using  $\mathcal{H}$ ;
7:   Fix  $\psi$ , obtain a instance sequence  $Q' = \{(\mathbf{x}, y, z)\}$ ;
8:    $\mathcal{Q} = \mathcal{Q} \cup Q'$ ;
9:   Update  $\theta$  by Equation 15 using  $\mathcal{Q}$ ;
10: end while
11: return  $\theta, \psi$ 

```

Algorithm 2 IDSS Inference

Input:

A instance sequence $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)}$;
The learned parameters θ^*, ψ^* ;

Output:

Predicted $\hat{y}^{(1)}, \dots, \hat{y}^{(T)}$;

```

1: Initialize  $\mathbf{s}^{(0)} = \mathbf{0}$ 
2: while  $t = 1, 2, \dots, T$  do
3:    $\mathbf{z}^{(t)} = \pi(\mathbf{s}^{(t-1)}; \psi^*)$ ;
4:    $\tilde{\mathbf{x}}^{(t)} = g(\mathbf{x}^{(t)}, \mathbf{z}^{(t)})$ ;
5:    $\hat{y}^{(t)} = \arg \max_y p(y|\tilde{\mathbf{x}}^{(t)}; \theta^*)$ ;
6:    $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \hat{\mathbf{p}}^{(t)})$ ;
7: end while
8: return  $\hat{y}^{(1)}, \hat{y}^{(2)}, \dots, \hat{y}^{(T)}$ 

```

start at the initial state $\mathbf{s}^{(0)} = \mathbf{0}$. For an incoming instance $\mathbf{x}^{(t)}$, we obtain the subset of sensors via the policy function:

$$\mathbf{z}^{(t)} = \pi(\mathbf{s}^{(t-1)}; \psi^*) \quad (17)$$

Then, only the required sensors are sampled their readings, which obtains an observed instance $\tilde{\mathbf{x}}^{(t)} = g(\mathbf{x}^{(t)}, \mathbf{z}^{(t)})$. After that, the classification model is used to make a prediction $\hat{y}^{(t)}$ for $\tilde{\mathbf{x}}^{(t)}$:

$$\hat{y}^{(t)} = \arg \max_y p(y|\tilde{\mathbf{x}}^{(t)}; \theta^*) \quad (18)$$

We obtain the instance-dependent state $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \hat{\mathbf{p}}^{(t)})$ which can be used to predict the required subset of sensors in the next timestep. The pseudo-code of IDSS inference process is provided in Algorithm 2.

Experiment

Experimental Setup

Datasets To evaluate the performance of the proposed method, comparative experiments are designed using three

real-world human activity datasets which are detailed in Table 1:

- **MHEALTH** (Banos et al. 2015): 10 subjects worn a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer on the right wrist and left ankle, and a 3-axis accelerometer on the chest.
- **PAMAP2** (Reiss and Stricker 2012): 8 subjects worn a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer on the chest and the dominant side’s wrist and ankle.
- **ActivityNet**: 8 subjects worn a 3-axis accelerometer on the chest, two wrists and two ankles.

Settings To capture the activities, 1-second sliding windows with none overlay are used to segment the data stream and all the numeric values are normalized into $[-1, 1]$. For each axis of sensors, we extract 13 statistic temporal domain features for $\phi(\tilde{\mathbf{x}})$, e.g., mean, variance, deviation. We use the Neural Network (NN) to build the classification model and the policy function in the MDP. Those features from unselected sensors are imputed by zeros. Using a sparse feature vector also improves classification efficiency at test time. Both the NNs are optimized by ADAM solvers. We conduct the experiments on leave-one-out cross-validation, where one of the subjects is selected for testing and the others for training. The iteration round $MaxIter$ for training is set to 10.

Baselines We test three baseline methods in comparison with IDSS, which are detailed as follows:

- **Fixed sensor set**: we report K results for this method, one for each sensor number budget. If selecting k out of K sensors, there are $\frac{K!}{k!}$ solutions. To speed up, we iteratively filter out a sensor in a greedy manner for evaluating the performance with k sensors. At each iteration, the sensor filtered out can be found as:

$$b^* = \arg \min_{b \in \mathcal{B}} \sum 1_{\mathcal{B} \setminus b}(y_t \neq \hat{y}_t) \quad (19)$$

where \mathcal{B} denotes the remaining sensor set.

- **Random**: the activity recognition model is trained with all the available sensors but a random subset of sensors is selected to make the classification at each testing timestep. In this setting, $\mathbb{E}_{t \sim T}[z_t] = K/2$.
- **MDP**: we implement a pure MDP to select sensors for each instance. The proposed Mutual DAGger is not used to enhance the learning process.

Evaluate the performances of IDSS

Since our goal is to balance the recognition accuracy and the sensor number, we define a comprehensive metric called *balance-index* to compare IDSS with other state-of-the-art methods:

$$balance - index = \frac{1}{T} \sum_{t=1}^T 1(y_t = \hat{y}_t) - \lambda \frac{\|z_t\|_1}{K} \quad (20)$$

Table 1: Dataset Description

Dataset	# Subject	# Sensor	Activity	
			# Number	Detail
MHEALTH	10	7 (60Hz)	12	Standing Still, Sitting and relaxing, Lying down, Walking, Climbing Stairs, Waist bends forward, Frontal elevation of arms, Knees bending, Cycling, Jogging, Running, Jump front & back
PAMAP2	8	9 (100Hz)	9	Lying, Sitting, Standing, Walking, Running, Cycling, Nordic Walking, Ascending Stairs, Descending Stairs
ActivityNet	8	5 (50Hz)	4	Standing, Sitting, Lying down, Walking

The results with $\lambda \in [0, 1]$ on the three datasets are shown in Table 2 where the bold one indicates the best performance. Comparing with other methods, IDSS shows better performance with regard to the balance-index, which indicates that it is better at balancing the recognition accuracy and sensor number. In other words, IDSS can use fewer sensors to obtain the same, if not better, accuracy performance.

- Comparing with fixed sensor set settings, IDSS performs better than the best-fixed sensor set which owing to that the sensor set is specified for each instance in IDSS and is better for classification.
- Comparing with other dynamic sensor settings, i.e., Random and MDP, IDSS enhances the two parts of parameters by proposed Mutual DAGger when employing Imitation Learning and is capable of dealing with various sensor subset situations, which results in a better performance.

Although not considering the sensor number when $\lambda = 0$, IDSS still performs the best than others, which shows that it is essential for each instance to select the optimal sensors. As λ increases, the advantage of IDSS is getting more outstanding. This is because with a large λ , IDSS selects fewer sensors without losing accuracy.

We further investigate IDSS by analyzing the two relationships: 1) λ and recognition accuracy; 2) λ and the average sensor number. As shown in Figure 1, λ has nearly no effect on the recognition accuracy, but the required sensors keeps reducing as the increasing of λ , since IDSS selects a appropriate sensor subset for each instance and makes an accurate classification. Since there is a limitation of the least number of sensors, the required sensor will not be reduced any longer when λ gets large enough.

To better illustrate the online process of IDSS, we present one of the dynamic sensor selection results in three datasets, which are shown in Figure 2. There are two y-axes in the figures: the left one denotes the sensor ID in the system and the right one denotes the activity ID. The x-axis denotes the timeline and the red dash line denotes the ongoing activities along the timeline. If the sensor is required in the timestep, the corresponding box will be painted black, or else be left blank. We can observe a trend that different activity may require different sensors. Those complex activities such as Frontal elevation of arms and Knees bending in MHEALTH require larger and various sensor subsets. What's more, the required sensor subset varies not only across activities but also across instances within one activity. In summary, IDSS manages reducing the average sensor number to 1.78, 2.58

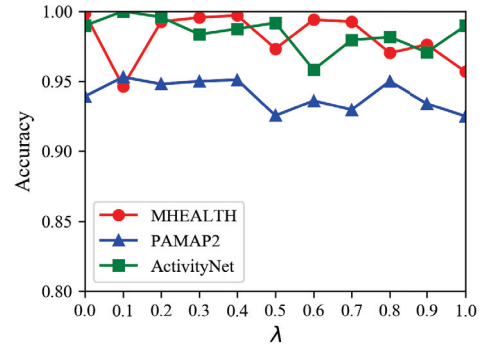
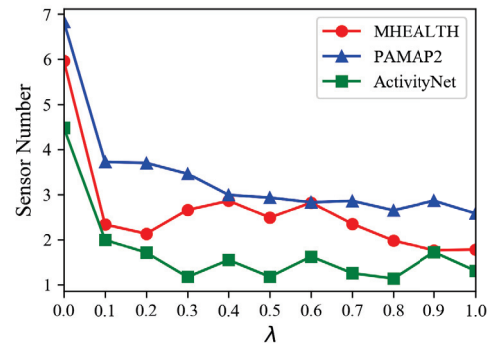
(a) λ and Accuracy(b) λ and Sensor Number

Figure 1: Two relationships on three datasets

and 1.63 from 7, 9, 5 sensors in MHEALTH, PAMAP2 and ActivityNet, correspondingly.

Conclusion & Future work

In this paper, we address an emerging issue of human activity recognition in resource-constrained environments by dynamically selecting a subset of sensors along with the instance sequence in order to balance recognition accuracy and sensor cost. We formulate this issue as a problem of minimizing an objective function regarding activity classification loss and required sensor number. Then we propose IDSS to solve the problem via an MDP whose policy is learned by directly exploiting the instance-wise continuous state using Imitation Learning. To learn two parts of param-

Table 2: The balance-index with respect to λ on MHEALTH, PAMAP2, ActivityNet

MHEALTH											
Method	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
7 sensors	0.9821	0.8821	0.7821	0.6821	0.5821	0.4821	0.3821	0.2821	0.1821	0.0821	-0.0179
6 sensors	0.9911	0.9054	0.8196	0.7339	0.6482	0.5625	0.4768	0.3911	0.3054	0.2196	0.1339
5 sensors	0.9360	0.8646	0.7932	0.7217	0.6503	0.5789	0.5074	0.4360	0.3646	0.2932	0.2217
4 sensors	0.9926	0.9354	0.8783	0.8211	0.7640	0.7068	0.6497	0.5926	0.5354	0.4783	0.4211
3 sensors	0.9970	0.9542	0.9113	0.8685	0.8256	0.7827	0.7399	0.6970	0.6542	0.6113	0.5685
2 sensors	0.8214	0.7929	0.7643	0.7357	0.7071	0.6786	0.6500	0.6214	0.5929	0.5643	0.5357
1 sensor	0.5000	0.4857	0.4714	0.4571	0.4429	0.4286	0.4143	0.4000	0.3857	0.3714	0.3571
Random	0.5506	0.5006	0.4506	0.4006	0.3506	0.3006	0.2506	0.2006	0.1506	0.1006	0.0506
MDP	0.4343	0.3471	0.4445	0.3189	0.3429	0.4764	0.3867	0.4264	0.3846	0.3778	0.3948
IDSS	0.9985	0.9488	0.9316	0.8816	0.8335	0.7955	0.7993	0.7576	0.7439	0.7493	0.7024
PAMAP2											
Method	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
9 sensors	0.9345	0.8345	0.7345	0.6345	0.5345	0.4345	0.3345	0.2345	0.1345	0.0345	-0.0655
8 sensors	0.8913	0.8025	0.7136	0.6247	0.5358	0.4469	0.3580	0.2691	0.1802	0.0913	0.0025
7 sensors	0.8854	0.8077	0.7299	0.6521	0.5743	0.4966	0.4188	0.3410	0.2632	0.1854	0.1077
6 sensors	0.9056	0.8389	0.7723	0.7056	0.6389	0.5723	0.5056	0.4389	0.3723	0.3056	0.2389
5 sensors	0.8741	0.8186	0.7630	0.7075	0.6519	0.5964	0.5408	0.4853	0.4297	0.3741	0.3186
4 sensors	0.8668	0.8223	0.7779	0.7334	0.6890	0.6445	0.6001	0.5557	0.5112	0.4668	0.4223
3 sensors	0.8476	0.8143	0.7809	0.7476	0.7143	0.6809	0.6476	0.6143	0.5809	0.5476	0.5143
2 sensors	0.8132	0.7910	0.7687	0.7465	0.7243	0.7021	0.6798	0.6576	0.6354	0.6132	0.5910
1 sensor	0.6549	0.6438	0.6326	0.6215	0.6104	0.5993	0.5882	0.5771	0.5660	0.5549	0.5438
Random	0.4572	0.4072	0.3572	0.3072	0.2572	0.2072	0.1572	0.1072	0.0572	0.0072	-0.0428
MDP	0.5200	0.4630	0.3894	0.4710	0.4431	0.4406	0.4302	0.5079	0.4112	0.4405	0.3671
IDSS	0.9393	0.9112	0.8661	0.8415	0.8182	0.7627	0.7476	0.7068	0.7147	0.6474	0.6380
ActivityNet											
Method	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
5 sensors	0.9958	0.8958	0.7958	0.6958	0.5958	0.4958	0.3958	0.2958	0.1958	0.0958	-0.0042
4 sensors	0.9958	0.9158	0.8358	0.7558	0.6758	0.5958	0.5158	0.4358	0.3558	0.2758	0.1958
3 sensors	0.9917	0.9317	0.8717	0.8117	0.7517	0.6917	0.6317	0.5717	0.5117	0.4517	0.3917
2 sensors	0.9917	0.9517	0.9117	0.8717	0.8317	0.7917	0.7517	0.7117	0.6717	0.6317	0.5917
1 sensor	0.7667	0.7467	0.7267	0.7067	0.6867	0.6667	0.6467	0.6267	0.6067	0.5867	0.5667
Random	0.7908	0.7408	0.6908	0.6408	0.5908	0.5408	0.4908	0.4408	0.3908	0.3408	0.2908
MDP	0.9855	0.7070	0.4405	0.6173	0.3998	0.3946	0.5028	0.2935	0.4186	0.3757	0.4266
IDSS	0.9958	0.9600	0.9273	0.8778	0.821	0.8225	0.7633	0.7465	0.6828	0.6603	0.6350

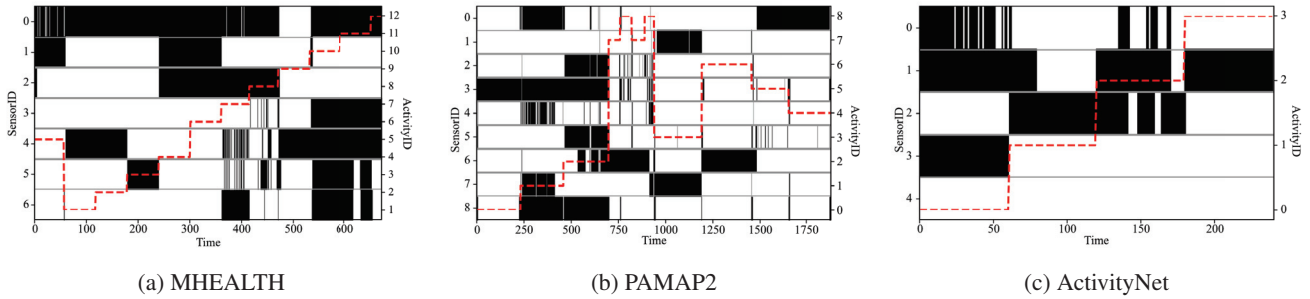


Figure 2: Example of IDSS on MHEALTH, PAMAP2 and ActivityNet

eters in IDSS, i.e., the classification model and the selection policy in MDP, we propose Mutual DAGger to make them enhance each other during the alternative learning process. We evaluated the performance on three real-world datasets. The experimental results show that IDSS significantly reduces the required sensors without losing accuracy and outperforms the state-of-the-art methods considering the combined measurement of recognition accuracy and sensor cost.

In future work, we will investigate applying IDSS to recognize more complex activities with more sensors and test it on more sensor modalities environment, such as combining the wearable sensors with those which are not body-worn.

Acknowledgement

This work was supported by National Key Research & Development Plan of China No.2017YFB1002802; Nat-

ural Science Foundation of China No.61972383 and No.61502456; Research & Development Plan in Key Field of Guangdong Province No.2019B010109001; and Beijing Municipal Science & Technology Commission No.Z171100000117013 and No.Z171100000117017.

References

- Aziz, O.; Robinovitch, S. N.; and Park, E. J. 2016. Identifying the number and location of body worn sensors to accurately classify walking, transferring and sedentary activities. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 5003–5006. IEEE.
- Banos, O.; Villalonga, C.; Garcia, R.; Saez, A.; Damas, M.; Holgado-Terriza, J. A.; Lee, S.; Pomares, H.; and Rojas, I. 2015. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online* 14(2):S6.
- Bellman, R. 1952. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America* 38(8):716.
- Bloom, V.; Argyriou, V.; and Makris, D. 2013. Dynamic feature selection for online action recognition. In *International Workshop on Human Behavior Understanding*, 64–76. Springer.
- Chen, Y.; Yang, X.; Chen, B.; Miao, C.; and Yu, H. 2017. Pdassist: Objective and quantified symptom assessment of parkinson’s disease via smartphone. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 939–945. IEEE.
- Cheng, W.; Erfani, S. M.; Zhang, R.; and Kotagiri, R. 2018. Learning datum-wise sampling frequency for energy-efficient human activity recognition. In *AAAI*.
- Dulac-Arnold, G.; Denoyer, L.; Preux, P.; and Gallinari, P. 2011. Datum-wise classification: a sequential approach to sparsity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 375–390. Springer.
- Ertuğrul, Ö. F., and Kaya, Y. 2017. Determining the optimal number of body-worn sensors for human activity recognition. *Soft Computing* 21(17):5053–5060.
- Gordon, D.; Czerny, J.; Miyaki, T.; and Beigl, M. 2012. Energy-efficient activity recognition using prediction. In *2012 16th International Symposium on Wearable Computers*, 29–36. IEEE.
- Gravina, R.; Alinia, P.; Ghasemzadeh, H.; and Fortino, G. 2017. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion* 35:68–80.
- He, H.; Daumé III, H.; and Eisner, J. 2012. Cost-sensitive dynamic feature selection. In *ICML Inferring Workshop*.
- Karayev, S.; Fritz, M. J.; and Darrell, T. 2013. Dynamic feature selection for classification on a budget. In *International Conference on Machine Learning (ICML): Workshop on Prediction with Sequential Models*.
- Kolamunna, H.; Hu, Y.; Perino, D.; Thilakarathna, K.; Makaroff, D.; Guan, X.; and Seneviratne, A. 2016. Afv: enabling application function virtualization and scheduling in wearable networks. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 981–991. ACM.
- Mazilu, S.; Blanke, U.; Roggen, D.; Tröster, G.; Gazit, E.; and Hausdorff, J. M. 2013. Engineers meet clinicians: augmenting parkinson’s disease patients to gather information for gait rehabilitation. In *Proceedings of the 4th Augmented Human International Conference*, 124–127. ACM.
- Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J. A.; Abbeel, P.; Peters, J.; et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics* 7(1-2):1–179.
- Reiss, A., and Stricker, D. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, 108–109. IEEE.
- Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635.
- Satsangi, Y.; Whiteson, S.; and Oliehoek, F. A. 2015. Exploiting submodular value functions for faster dynamic sensor selection. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Shen, X., and Varshney, P. K. 2013. Sensor selection based on generalized information gain for target tracking in large sensor networks. *IEEE Transactions on Signal Processing* 62(2):363–375.
- Spaan, M. T., and Lima, P. U. 2009. A decision-theoretic approach to dynamic sensor selection in camera networks. In *Nineteenth International Conference on Automated Planning and Scheduling*.
- Tang, L. M.; Meyer, J.; Epstein, D. A.; Bragg, K.; Engelen, L.; Bauman, A.; and Kay, J. 2018. Defining adherence: Making sense of physical activity tracker data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2(1):37.
- Trapeznikov, K., and Saligrama, V. 2013. Supervised sequential classification under budget constraints. In *Artificial Intelligence and Statistics*, 581–589.
- Wang, A.; Lin, F.; Jin, Z.; and Xu, W. 2016. A configurable energy-efficient compressed sensing architecture with its application on body sensor networks. *IEEE Transactions on Industrial Informatics* 12(1):15–27.
- Yu, H.; Yang, X.; Chen, Y.; and Liu, J. 2015. strdoctor: Indicate stroke for elderly through body sensing game. In *2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing*, 360–363. IEEE.
- Zappi, P.; Lombriser, C.; Stiefmeier, T.; Farella, E.; Roggen, D.; Benini, L.; and Tröster, G. 2008. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *European Conference on Wireless Sensor Networks*, 17–33. Springer.