2018 IEEE 38th International Conference on Distributed Computing Systems

Towards Personalized Learning in Mobile Sensing Systems

Wenjun Jiang[1], Qi Li[2], Lu Su[1*], Chenglin Miao[1], Quanquan Gu[3], and Wenyao Xu[1]
[1] *State University of New York at Buffalo,* [2] *University of Illinois at Urbana-Champaign,* [3] *University of Virginia*
Email: [1] {*wenjunji, lusu, cmiao, wenyaoxu*}*@buffalo.edu,* [2] *qili5@illinois.edu,* [3] *qg5w@virginia.edu*

***Abstract*—Nowadays, mobile devices have become an important part of our daily life. Numerous mobile sensing applications are enabled by various mobile platforms, which leverage machine learning techniques to detect or classify the events of interest such as human activities and health conditions. To achieve this, each user is required to provide a considerable amount of training samples. However, in practice, a large portion of the users may provide only a few or even zero labels, due to various reasons such as privacy concern or simply laziness. A straightforward solution to this problem is to gather the data of all the users in a central database, and train a global classifier from the combined data. Such global classifier, however, may not work well since *it ignores the variety in different users' data*. To address this challenge, we propose PLOS, a Personalized Learning framework for mObile Sensing applications. PLOS can *jointly model the commonness shared among the users as well as the differences between them*, which are inferred from both the label information and the underlying structures of individual data. We further develop the distributed PLOS where the raw data of the users are locally processed so that the users only need to send model parameters to the server. Through extensive experiments on both synthetic data and real mobile sensing systems, we show that the proposed PLOS framework is scalable and efficient in energy, computation, and communication costs, and can achieve more accurate classification results compared with the baseline methods.**

***Keywords*-personalized learning; mobile sensing; distributed machine learning;**

I. Introduction

With the rapid development of sensing, communicating, and computing technologies, the pervasive mobile smart devices have revolutionized our daily life with countless mobile sensing applications [1]–[9], such as health care [1], smart home [3], assisted driving [7], and intelligent shopping [8], which fundamentally change the ways in which we interact with the physical world [4].

In mobile sensing systems, machine learning techniques are widely applied so that the sensing tasks can be fulfilled in an automatic and adaptive manner. For example, consider the task of activity recognition, a classifier can be trained based on the data collected by mobile devices. When a new motion is detected, the classifier can automatically recognize the corresponding activity. Ideally, the classifiers should be trained based on the individual users' data. This requires that each user provides sufficient amount of training samples to the machine learning algorithms. However, in real-life mobile sensing systems, *a large portion of the users may provide only a few or even zero labels*, due to various reasons such as the concern of privacy leakage, the consumption of time and other resources, or simply carelessness and laziness. In such cases, it is hard for the classifier to achieve satisfactory accuracy.

A naive solution to handle insufficient training data for individual users is to collect many users' labeled information and put them into a centralized training pool. Then a global classifier can be trained on the combined data. However, this solution has some major drawbacks in real-life applications. 1) To combine the label information from multiple users, it requires the users upload their raw data to a central server. In many applications, due to the communication/energy constraints as well as the privacy concern, it is not feasible. 2) Even if all the data can be delivered to the server, the global classifier trained upon the combined data may not be suitable for every user. For example, a classifier learned from the activity data collected from a group in which the majority of the people are adults cannot perform well when being applied to recognize the activities of kids, disabled, and senior persons. In real world, for many mobile sensing tasks, such as health condition monitoring, activity recognition, facial expression recognition, and handwriting recognition, the learning process needs to be personalized to improve user experience, since different users may demonstrate different patterns on the same tasks. Thus, applying the same classifier on different users will not be able to achieve satisfactory performance due to *the ignorance of individual differences*.

To address the aforementioned challenges, we propose a Personalized Learning in mObile Sensing (**PLOS**) framework, which *jointly captures the commonness and the differences of individual users simultaneously*. On one hand, by modeling the commonness, PLOS enables the users to share knowledge with each other, and thus can benefit the users with insufficient or even zero training data. On the other hand, by modeling the differences, PLOS characterizes the structures of individual data, and thus can benefit the users with unique data patterns. To achieve this, PLOS jointly learns a global classifier for all the users and a bias for each individual. By integrating the global classifier with individual bias, PLOS produces a personalized classifier for each user that can maximize the margin between his/her classes. In a word, PLOS makes a full use of all the available information from a large population, from label information to the underlying data structure, and calibrates such informa-

[*] L. Su is the corresponding author.

2575-8411/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDCS.2018.00040

IEEE computer society

tion into personalized knowledge for each individual. Such knowledge personalization from population to individual is the key achievement of this work.

Moreover, the proposed PLOS framework can be implemented in a distributed and parallel manner, and thus can address a series of issues such as privacy and computation/communication efficiency. In the distributed PLOS framework, the raw data are locally processed on the smart device of each user instead of the server, and the users only communicate with the server. The benefit of such design is multi-fold. Firstly, the privacy of each user is protected, as he/she does not have to share his/her data with the server or other users. Secondly, the cost of communication is greatly reduced, since during the learning process, only the model parameters instead of raw data are exchanged between each user and the server. Last but most importantly, distributed PLOS does not sacrifice the learning performance.

In summary, the main contributions of this paper are:

- We address the challenge of personalized learning in mobile sensing systems, when the users provide insufficient or even no label information to the system. The proposed PLOS system jointly models users' commonness and uniqueness simultaneously to improve learning accuracy.
- The proposed PLOS is a distributed framework that has multi-fold benefits in real-life mobile sensing systems. The users only communicate with the server with a considerably small amount of messages. Therefore, the privacy issue as well as the communication, energy, and time cost are all addressed.
- The PLOS framework is tested on a real mobile sensing system, and the experimental results demonstrate its superior performance in both accuracy and efficiency.

In the rest of the paper, we first discuss the related work in Section II. Then we formally define the problem and present an overview of the PLOS system in Section III. The proposed framework is introduced in Section IV and Section V and evaluated in Section VI. Finally, we conclude the paper and discuss some future work in Section VII.

II. Related Work

Personalized Learning. Some previous work has also studied the personalized learning problem. [10]–[21], most of them [10]–[17] cannot be applied to the scenarios where not all the users have label information. But our proposed framework can relax this assumption and incorporate users who do not have any labeled data. In [20], the authors utilize social network as a measure of similarity among users and then group the users into cliques according to their similarities. In contrast, we do not assume the availability of social network information. In [18], [19], the authors assume that the users can share their data with some of the other users during the training process. But in our setting, sharing personal data is not allowed for the sake of privacy preserving. Furthermore, our proposed method jointly learns the commonness among the users and the difference between the users to solve the challenge that users provide insufficient or even no label information to the system, which are not considered in previous works.

Transfer Learning. One related area of the proposed method is transfer learning. Transfer learning refers to a branch of machine learning tasks, where the knowledge is learned from one problem domain, called source domain, but applied to another problem domain, called target domain [22]. Transfer learning has been applied in many applications [23]. In [24]–[27], the authors apply Teacher/Learner transfer learning model, where the classifier trained on the teacher sensor provides labels to the learner sensor that is deployed on the same object as the teacher (e.g., the teacher sensor and learner sensor are worn on the hand and leg of the same person). In [3], [28]–[30], the domain adaptation techniques were applied. After the adaptation, the knowledge of the sensor on the source object can be used by the sensor deployed on the target object. Compared with these existing methods, where the knowledge is one-way transferred from one or multiple sources to a single target, the proposed method enables knowledge sharing among all the users so that they can mutually benefit each other.

Multi-Task Learning. Another related area is multi-task learning. The goal of multi-task learning is to conduct multiple similar learning tasks simultaneously. To achieve this, the multi-task learning models need to capture the similarities among different tasks. One line of multi-task learning work handles the similarities among tasks from the probabilistic point of view [11], [12], [31], [32], while another line of work uses optimization techniques to model the similarities [10], [15], [16], [33]–[35]. No matter how they model the tasks in the aforementioned literature, there is one requirement in common: all the tasks need to provide label information. Different from existing multi-task learning models, the proposed method relaxes this requirement and thus can be applied to more general scenarios.

Distributed Machine Learning. Our work also relates to distributed machine learning, which tries to learn from data that are stored in distributed databases, and are costly or even infeasible to be transmitted over network due to their volumes or sensitivity [36]. Studies of distributed machine learning include distributed classification and regression [37]–[44], distributed clustering [45]–[49], and ensemble learning [50], [51]. Though following a distributed design, none of the above work explores personalized learning for different individuals.

III. Problem Formulation and System Overview

Considering a mobile sensing task where T users, indexed as $t = 1, \cdots, T$, cooperate together to train classifiers for the same purpose, for example, to classify their activities such as walking, jogging, laying. The data are collected from individual users. Using the activity recognition example, the data are the sensory data (accelerometer, gyroscope, etc) collected from their mobile devices. Among all the users, some do not provide any label, while others are willing to manually label part of their samples. The data provided by the users may follow different distributions. That is, different persons may demonstrate different patterns in their activities. Thus, instead of learning a common decision boundary (i.e., classifier) for all the users, our task is to learn a personalized decision boundary for each user.

Mathematically, we use subscript t to index the notations with respect to user t, whose data points are denoted as $\{\mathbf{x}_{1_t}, \mathbf{x}_{2_t}, \cdots, \mathbf{x}_{m_t}\}$. Without loss of generality, we assume that the first l_t samples in $\mathbf{x}_t$ are labeled, denoted as $\{(\mathbf{x}_{1_t}, y_{1_t}), (\mathbf{x}_{2_t}, y_{2_t}), \cdots, (\mathbf{x}_{l_t}, y_{l_t})\}$, and the rest $m_t - l_t$ samples are unlabeled, denoted as $\{\mathbf{x}_{l_t+1}, \mathbf{x}_{l_t+2}, \cdots, \mathbf{x}_{m_t}\}$. Note that if a user does not provide labels, $l_t = 0$. The decision boundary we want to learn is a hyperplane[1] $f_t(\mathbf{x}) = \mathbf{w}_t \cdot \mathbf{x}$.

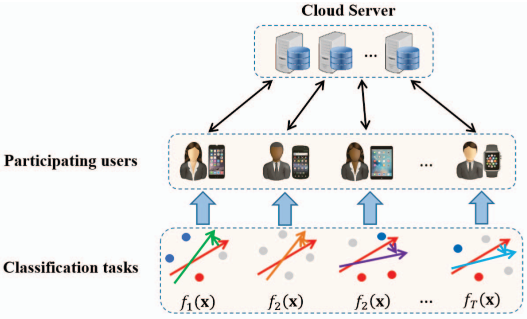


Figure 1. Overview of the **PLOS** System

Figure 1 illustrates the architecture of the proposed PLOS system. Considering the issues of privacy concern as well as the bandwidth, computation, and communication cost, the proposed PLOS framework follows a distributed design where each user locally conducts calculations on the raw data, and only communicates to the server with intermediate model parameters. Specifically, each user has a local dataset of raw sensory data, some users manually label part of their data to positive (red dots in Figure 1) or negative (blue dots), while the others do not provide any label information (gray dots). The goal of each user is to train a personalized classifier (arrow lines with green, orange, purple, and blue color) to classify data into positive or negative. Since all the users are doing the same activities, their classifiers share some commonness. Meanwhile, since they demonstrate different patterns in their activities, they have some uniqueness in their data. The PLOS system models the commonness among the users with a global classifier (red arrow line), and allows each individual classifier deviate the global classifier a bit to reflect their uniqueness. Then the PLOS system jointly learns the classifiers of each user in a distributed way, in which the information transmitted between the server and the users are the parameters of the local classifiers and the global classifier, instead of the raw data of the users, thus it can provide more privacy protection for the users and saves the computation and communication cost. In the following sections, we first introduce how to address the personalized learning in the centralized scenario (Section IV), where all the users need to upload their data to the central server. Then we extend it to a distributed approach (Section V), in which the users do not upload their personal data.

IV. Centralized PLOS

In this section, we describe the centralized PLOS method, which tackles the problem of personalized learning for mobile sensing tasks. We model this problem in an optimization framework where a personalized hyperplane is learned based on not only a user's individual data but also the knowledge "borrowed" from other users.

A. *The PLOS framework*

The proposed PLOS framework inherits the spirit of Support Vector Machine (SVM), one of the most widely used classification algorithms. The key idea of SVM algorithm, which is applied in the design of PLOS, is to find a hyperplane that can maximize the margin (i.e., the distance from the hyperplane to the nearest data point on each side). For simplicity, the labels are assumed to be in $\{-1, 1\}$.

Mathematically, SVM is formulated as follows:

$$\begin{aligned} \min_{\mathbf{w}, \xi_i \geq 0} \quad & \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 - \xi_i, \forall i = 1, \cdots, m, \end{aligned} \tag{1}$$

where the slack variables ξ_i are used to allow some degree of slackness in the constraint so that the algorithm is not oversensitive to possible outliers.

Though sharing the same spirit, SVM algorithm is not suitable in our setting due to the following reasons. 1) Some users do not provide labels for their data. For these users, SVM cannot be applied. For those who provide labels, the labeled samples may also be too sparse to train a good classifier. 2) If we use the data from all users to train a single global SVM classifier, there may be enough training samples, but the different patterns lying in individual users' data are disregarded. Consequently, the global classifier may not be able to accurately classify each individual user's data.

To conquer these challenges, the idea of the proposed model is to jointly consider the labeled and unlabeled data from all users, identify their commonness, and at the same time capture their individual characteristics. On one hand, the commonness is shared among all users, so the information from different users can be unified to help each

[1] In this format, the hyperplane will have to go through the origin. Generalization can be easily achieved by adding another dimension on $\mathbf{x}_{i_t}$ with constant 1. In such way, the corresponding dimension in $\mathbf{w}_t$ is the bias of the hyperplane.

other, especially the users who have no labeled data. On the other hand, individual users may behave differently, so the personalized learning can improve the classification performance by characterizing the difference among users.

We model the commonness among all individual hyperplanes $\mathbf{w}_t$ as a global hyperplane $\mathbf{w}_0$. And we let the first n of the total T datasets have labels. Inspired by the idea of maximum margin clustering [52], we formulate our framework as the following two-layer optimization problem:

$$\begin{aligned}\min_{\{y_{i_t}\}_{\forall i=l_t+1,\cdots,m_t}} \min_{\mathbf{w}_0,\mathbf{w}_t,\xi_{i_t}\geq 0} &\Big\{||\mathbf{w}_0||^2 + \frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{w}_t-\mathbf{w}_0||^2 \\ &+\sum_{t=1}^{T}(\frac{C_l}{m_t}\sum_{i=1}^{l_t}\xi_{i_t} + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t}\xi_{i_t})\Big\} \\ \text{s.t.}\quad &\forall t=1,\cdots,T: \\ &y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t}) \geq 1-\xi_{i_t}, \forall i=1,\cdots,m_t. \end{aligned} \qquad (2)$$

The inner optimization tries to find the best hyperplanes given the current label assignments $y_{i_t}, \forall i=1,\cdots,m_t$. The outer optimization tries to find the best label assignments for the unlabeled data $y_{i_t}, \forall i=l_t+1,\cdots,m_t$. Note that y_{i_t} is a constant (i.e., user-provided label) if $i \leqslant l_t$.

In the objective function of the inner optimization problem, there are three parts needed to be minimized. The first part $||\mathbf{w}_0||^2$ maximizes the margin of the global hyperplane $\frac{2}{||\mathbf{w}_0||}$. The second part $\frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{w}_t-\mathbf{w}_0||^2$ minimizes the difference between the global hyperplane and those of the users. And the third part $\sum_{t=1}^{T}(\frac{C_l}{m_t}\sum_{i=1}^{l_t}\xi_{i_t} + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t}\xi_{i_t})$ indicates the classification errors given the current label assignments. The slack variable ξ_{i_t} can be derived from the constraint: $\xi_{i_t} \geq \max\{0, 1-y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\}$. Therefore, minimizing ξ_{i_t} is equivalent to minimizing the error of the model on $\mathbf{x}_{i_t}$.

There are three predefined parameters in the objective function, namely λ, C_l, and C_u. λ is a positive regularization parameter that controls how much $\mathbf{w}_t$ can differ from the global hyperplane $\mathbf{w}_0$. When λ is large, it will give more penalties on $||\mathbf{w}_t-\mathbf{w}_0||^2$, so the hyperplanes for the users will be more similar to each other. On the other hand, when λ is small, the hyperplanes will rely more on the individual users' data. C_l and C_u together control the weight of ξ_{i_t}, the learning errors. Moreover, C_l and C_u also control the importance of the labeled data and unlabeled data respectively.

One difficulty of solving the problem (2) comes from the fact that we have to minimize the objective function with respect to all label assignments on $y_{i_t}, \forall i=l_t+1,\cdots,m_t$. In fact, for any given $\mathbf{w}_0$ and $\mathbf{w}_t$, the optimal label assignments are $y_{i_t} = \text{sign}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})$. It can be easily derived from the fact that these assignments give the minimum classification errors, which implies the minimum ξ_{i_t}, and thus the optimal objective value. Therefore, the outer optimization can be merged into the inner optimization, and $y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})$ can be replaced by $|\mathbf{w}_t\cdot\mathbf{x}_{i_t}|$. Then, the optimization problem (2) is equivalent to:

$$\begin{aligned}\min_{\mathbf{w}_0,\mathbf{w}_t,\xi_{i_t}\geq 0} &\Big\{||\mathbf{w}_0||^2 + \frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{w}_t-\mathbf{w}_0||^2 \\ &+\sum_{t=1}^{T}(\frac{C_l}{m_t}\sum_{i=1}^{l_t}\xi_{i_t} + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t}\xi_{i_t})\Big\} \\ \text{s.t.}\quad &\forall t=1,\cdots,T: \\ &y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t}) \geq 1-\xi_{i_t}, \forall i=1,\cdots,l_t, \\ &|\mathbf{w}_t\cdot\mathbf{x}_{i_t}| \geq 1-\xi_{i_t}, \forall i=l_t+1,\cdots,m_t. \end{aligned} \qquad (3)$$

In the above optimization problem, the number of slack variables ξ_{i_t} are as many as the number of data samples. To reduce the number of slack variables, we reformulate the optimization problem to the following:

$$\begin{aligned}\min_{\mathbf{w}_0,\mathbf{w}_t,\xi_t\geq 0}\quad &\Big\{||\mathbf{w}_0||^2 + \frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{w}_t-\mathbf{w}_0||^2 + \sum_{t=1}^{T}\xi_t\Big\} \\ \text{s.t.}\quad &\forall t=1,\cdots,T, \forall \mathbf{c}_t \in \{0,1\}^{m_t}: \\ &\frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t}c_{i_t}y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t}) + C_u\sum_{i=l_t+1}^{m_t}c_{i_t}\cdot|\mathbf{w}_t\cdot\mathbf{x}_{i_t}|\Big\} \\ &\geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t}c_{i_t} + C_u\sum_{i=l_t+1}^{m_t}c_{i_t}\Big\} - \xi_t, \end{aligned} \qquad (4)$$

where each $\mathbf{c}_t = (c_{1_t},\cdots,c_{m_t}) \in \{0,1\}^{m_t}$ selects a subset of the constraints in problem (3) to add them up. The equivalence is established since problem (3) and (4) have the same solution. For any given $\mathbf{w}_0$, $\mathbf{w}_t$, the ξ_{i_t} in problem (3) can be optimized individually. The optimum of (3) is achieved when

$$\xi_{i_t}^* = \begin{cases} \max\{0, 1-y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\} & i=1,\cdots,l_t; \\ \max\{0, 1-|\mathbf{w}_t\cdot\mathbf{x}_{i_t}|\} & i=l_t+1,\cdots,m_t. \end{cases} \qquad (5)$$

Similarly, the optimal ξ_t in problem (4) is

$$\begin{aligned}\xi_t^* = \max_{\mathbf{c_t}\in\{0,1\}^{m_t}} &\Big\{\frac{C_l}{m_t}\sum_{i=1}^{l_t}c_{i_t}\Big[1-y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\Big] \\ &+\frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t}c_{i_t}\Big[1-|\mathbf{w}_t\cdot\mathbf{x}_{i_t}|\Big]\Big\}. \end{aligned} \qquad (6)$$

It is clear that $\xi_t^* = \frac{C_l}{m_t}\sum_{i=1}^{l_t}\xi_{i_t}^* + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t}\xi_{i_t}^*$. Thus, the objective function of problem (3) and problem (4) have the same value. Hence, we conclude that problem (3) and problem (4) are equivalent.

We can simplify the optimization problem (4) through feature mapping and the kernel as described in [33]. Specifically, we define that

$$\Phi(\mathbf{x}_{i_t}) = (\frac{\mathbf{x}_{i_t}}{\sqrt{T/\lambda}}, \underbrace{\mathbf{0},\cdots,\mathbf{0}}_{t-1}, \mathbf{x}_{i_t}, \underbrace{\mathbf{0},\cdots,\mathbf{0}}_{T-t}), \qquad (7)$$

where $\mathbf{0}$ is a zero vector with the same dimension as $\mathbf{x}_{i_t}$. We also define that

$$\mathbf{w}' = (\sqrt{T/\lambda}\mathbf{w}_0, \mathbf{w}_1-\mathbf{w}_0, \cdots, \mathbf{w}_T-\mathbf{w}_0). \qquad (8)$$

Then we have $\mathbf{w}_t\cdot\mathbf{x}_{i_t} = \mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})$ and $||\mathbf{w}_0||^2 +$

$\frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{w}_t - \mathbf{w}_0||^2 = \frac{\lambda}{T}||\mathbf{w}'||^2$. Substituting them into (4) and multiplying the objective function by a constant $\frac{T}{2\lambda}$, we can reformulate the optimization problem to contain only one hyperplane $\mathbf{w}'$ as follows:

$$\begin{aligned}
\min_{\mathbf{w}',\xi_t\geq 0} \quad & \Big\{\frac{1}{2}||\mathbf{w}'||^2 + \frac{T}{2\lambda}\sum_{t=1}^{T}\xi_t\Big\} \\
\text{s.t.} \quad & \forall t = 1,\cdots,T, \forall \mathbf{c}_t \in \{0,1\}^{m_t}: \\
& \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})) + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot|\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})|\Big\} \\
& \geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\} - \xi_t. \qquad (9)
\end{aligned}$$

B. Optimization via concave-convex process

Problem (9) is non-convex since the constraints are non-convex. However, if the constraints can be expressed as the difference of two convex functions less than or equal to a constant, the concave-convex process (CCCP) can be used to solve this problem [53]. CCCP is an iterative process. The main idea is that at each iteration, the constraints are approximated by convex functions localized at the previous estimations (or an initialization if this is the first iteration), and then solve the approximated problem and update the estimations. To construct the convex approximation, the second convex function is replaced by its first-order Taylor expansion at the previous estimation.

Specifically in problem (9), the constraints can be expressed as the difference of the following two convex functions less than or equal to a constant: $-\xi_t - \frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t}))$ and $\frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot|\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})|$. If the previous estimation is $(\mathbf{w}'^{(k)}, \xi_t^{(k)})$, then $|\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})|$ in the second convex function is replaced by its first-order Taylor expansion at $(\mathbf{w}'^{(k)}, \xi_t^{(k)})$ as follows:

$$\begin{aligned}
&|\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t})| + \text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}' - \mathbf{w}'^{(k)})\cdot\Phi(\mathbf{x}_{i_t}) \\
&= \text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})). \qquad (10)
\end{aligned}$$

Plugging Equation (10) into the problem (9), we get:

$$\begin{aligned}
\min_{\mathbf{w}',\xi_t\geq 0} \quad & \Big\{\frac{1}{2}||\mathbf{w}'||^2 + \frac{T}{2\lambda}\sum_{t=1}^{T}\xi_t\Big\} \\
\text{s.t.} \quad & \forall t = 1,\cdots,T, \forall \mathbf{c}_t \in \{0,1\}^{m_t}: \\
& \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})) \\
& + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot\text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t}))\Big\} \\
& \geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\} - \xi_t. \qquad (11)
\end{aligned}$$

Problem (11) is a convex optimization problem and we can get a new estimation $(\mathbf{w}'^{(k+1)}, \xi_t^{(k+1)})$ by solving it. The CCCP process will monotonically decrease the objective value [54], which is also bounded. So the convergence is guaranteed.

Problem (11) is still difficult to solve because there are as many as $\sum_{t=1}^{T} 2^{m_t}$ constraints which come from the 2^{m_t} possible assignments of vector $\mathbf{c}_t$. In order to solve (11) efficiently, we apply the cutting plane algorithm [55]. The main idea of the cutting plane algorithm is to construct successively tighter relaxations to the problem until getting a sufficiently accurate solution. Specifically, we keep a constraint subset Ω_t (empty set as initialization). At each step, we solve the problem with respect to $\forall \mathbf{c}_t \in \Omega_t, t = 1,\cdots,T$. Then we find the most violated constraint and add it to the constraint subset Ω_t. With the growth of Ω_t, a successively tightened approximation of the problem (11) is constructed. The algorithm stops when the most violated constraint is violated by no more than ϵ.

Mathematically, the optimization problem in each step of the cutting plane algorithm has the following form:

$$\begin{aligned}
\min_{\mathbf{w}',\xi_t\geq 0} \quad & \Big\{\frac{1}{2}||\mathbf{w}'||^2 + \frac{T}{2\lambda}\sum_{t=1}^{T}\xi_t\Big\} \\
\text{s.t.} \quad & \forall t = 1,\cdots,T, \forall \mathbf{c}_t \in \Omega_t: \\
& \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})) \\
& + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot\text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t}))\Big\} \\
& \geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\} - \xi_t. \qquad (12)
\end{aligned}$$

The most violated constraint is defined as the constraint $\mathbf{c}_t$ that produces the largest ξ_t, i.e.,

$$\begin{aligned}
\mathbf{c}_t = \underset{\mathbf{c}_t\in\{0,1\}^{m_t}}{\text{argmax}} & \Big\{\frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t}\Big[1 - y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t}))\Big] \qquad (13) \\
& + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t}\Big[1 - \text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t}))\Big]\Big\}.
\end{aligned}$$

Since each c_{i_t} can be optimized individually, it is easy to find that the most violated constraint can be chosen as:

$$c_{i_t} = \begin{cases} 1 & y_{i_t}(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})) < 1, \forall i = 1,\cdots,l_t; \\ 1 & \text{sign}(\mathbf{w}'^{(k)}_t\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})) < 1, \\ & \forall i = l_t+1,\cdots,m_t; \\ 0 & otherwise. \end{cases} \qquad (14)$$

If the solution ($\mathbf{w}'^*$, ξ_t^*) of the problem (12) and most violated constraint $\mathbf{c}_t$ of all the tasks satisfy the following inequality:

$$\begin{aligned}
\xi_t^* + \epsilon \geq & \frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t}\Big[1 - y_{i_t}(\mathbf{w}'^*\cdot\Phi(\mathbf{x}_{i_t}))\Big] \qquad (15) \\
& + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\{1 - \text{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))(\mathbf{w}'^*\cdot\Phi(\mathbf{x}_{i_t}))\Big\},
\end{aligned}$$

it means ($\mathbf{w}'^*$, $\xi_t^* + \epsilon$) is a feasible solution to the problem (11). Then the cutting plane algorithm can stop.

We can optimize the dual form of problem (12) through quadratic programming. The primal problem (12) is a convex problem, so Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for optimization. The dual optimization problem is as followed:

$$\max_{\gamma_{k_t} \geq 0} \quad -\frac{1}{2}||\sum_{t=1}^{T}\sum_{k=1}^{|\Omega_t|}\gamma_{k_t}\hat{\mathbf{z}}_{k_t}||^2 + \sum_{t=1}^{T}\sum_{k=1}^{|\Omega_t|}\gamma_{k_t}\hat{c}_{k_t}$$

$$\text{s.t.} \quad \sum_{k=1}^{|\Omega_t|}\gamma_{k_t} \leq \frac{T}{2\lambda}, \tag{16}$$

where

$$\hat{\mathbf{z}}_{k_t} = \frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t k_t} y_{i_t}\Phi(\mathbf{x}_{i_t}) + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t k_t}\cdot \operatorname{sign}(\mathbf{w}'^{(k)}\cdot\Phi(\mathbf{x}_{i_t}))\Phi(\mathbf{x}_{i_t}), \tag{17}$$

$$\hat{c}_{k_t} = \frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t k_t} + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t k_t}. \tag{18}$$

This is a quadratic programming (QP) problem with variables $(\gamma_{k_t})_{1\times\sum_{t=1}^{T}|\Omega_t|}$. So it can be solved via the standard form. We omit the details of derivations in this paper. After solving the QP problem, we get the solution $\mathbf{w}'^*$ and therefore can calculate the slack variable ξ_t^* and the optimal value of the objective function $L = \frac{1}{2}||\mathbf{w}'||^2 + \frac{T}{2\lambda}\sum_{t=1}^{T}\xi_t$ in the primal problem (12).

The algorithm flow is summarized in Algorithm 1.

Algorithm 1 : Centralized PLOS

```
1. Initialization. Set $\lambda, C_l, C_u, \epsilon, \mathbf{w}'^{(0)}$;
2. The CCCP process. Apply first-order Taylor expansion to $|\mathbf{w}'\cdot\Phi(\mathbf{x}_{i_t})|$ to get the convex optimization (11);
3. The cutting plane process. Initialize $\Omega_t = \phi$;
4. Solve problem (12) via its dual problem (16) to get $\mathbf{w}'^*$ and the primal form (12) to get $\xi_t^*$ and $L^*$;
5. Calculate the most violated constraint $\mathbf{c}_t$ of each user via equation (14).
6. If $\forall t$, $\mathbf{c}_t$ violate $\xi_t^*$ no more than $\epsilon$, goto step 7; otherwise, $\Omega_t = \Omega_t \cup \mathbf{c}_t$ and goto step 4;
7. If the difference between two consecutive $L^*$ is less than a given threshold, output $\mathbf{w}'^*$; otherwise goto step 2.
```

V. Distributed PLOS

The centralized PLOS method requires all users upload their data to the central server, so that the server can conduct all the computations. However, in real-life scenarios, the centralized methods may be inapplicable due to various reasons, such as privacy issues, bandwidth/energy constraints, and computation time requirement. Therefore, we further extend the centralized PLOS into a distributed method, where each user can locally conduct computations on the raw data, and only upload the intermediate results to the server. Compared with the centralized PLOS, the proposed distributed PLOS method can better handle the aforementioned problems.

The distributed PLOS method is developed based on the alternating direction method of multipliers (ADMM) framework [56]. ADMM is designed to solve a convex problem with equality constraints and it splits the variables into two parts x and z to exploit the decomposability of the variables.

In order to apply ADMM to solve the optimization problem (4), we firstly use the CCCP approach to convert it into a convex problem and add $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$ as a constraint to the problem. Thus, the problem is transformed as follows:

$$\min_{\mathbf{w}_0,\mathbf{v}_t,\mathbf{w}_t,\xi_t\geq 0} \quad \Big\{||\mathbf{w}_0||^2 + \frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{v}_t||^2 + \sum_{t=1}^{T}\xi_t\Big\}$$

$$\text{s.t.} \quad \forall t = 1,\cdots,T, \forall \mathbf{c}_t \in \{0,1\}^{m_t}:$$

$$\frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t}) + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot\operatorname{sign}(\mathbf{w}_t^{(k)}\cdot\mathbf{x}_{i_t})(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\Big\} \geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\} - \xi_t,$$

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t. \tag{19}$$

As an equivalent transformation, if we replace the slack variables ξ_t's by their optima

$$\xi_t^* = \max_{\mathbf{c_t}\in\{0,1\}^{m_t}}\Big\{\frac{C_l}{m_t}\sum_{i=1}^{l_t} c_{i_t}(1 - y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t})) + \frac{C_u}{m_t}\sum_{i=l_t+1}^{m_t} c_{i_t}(1 - \operatorname{sign}(\mathbf{w}_t^{(k)}\cdot\mathbf{x}_{i_t})(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\Big\}, \tag{20}$$

we can move the inequality constraint to the objective function and remove the slack variables ξ_t. Then the rest variables are partitioned as $x = [\mathbf{w}_1^T, \mathbf{v}_1^T\cdots, \mathbf{w}_t^T, \mathbf{v}_t^T]^T$, and $z = \mathbf{w}_0$. Then, we can define function $g(z) = ||\mathbf{w}_0||^2$ and $f(x)$ the rest in the objective function (19). Obviously, $g(z)$ and $f(x)$ are convex. The objective function in (19) is decomposable with respect to each variable in x. The augmented Lagrangian [57] is then

$$L_\rho(x,z,y) = f(x) + g(z) + (\rho/2)\sum_{t=1}^{T}||\mathbf{w}_t - \mathbf{v}_t - \mathbf{w}_0 + \mathbf{u}_t||^2, \tag{21}$$

where $u = (\mathbf{u}_1,\cdots,\mathbf{u}_T)$ are the scaled dual variables.

The calculation of x can be decomposed. Specifically, the update of $\mathbf{w}_t$ and $\mathbf{v}_t$ can be locally conducted by user t without communicating with any other user. Each user solves the following QP problem using the cutting plane algorithm with his/her local constraint subset Ω_t.

$$\min_{\mathbf{w}_t,\mathbf{v}_t,\xi_t\geq 0} \quad \Big\{\xi_t + \frac{\lambda}{T}||\mathbf{v}_t||^2 + (\rho/2)||\mathbf{w}_t - \mathbf{w}_0 - \mathbf{v}_t + \mathbf{u}_t||^2\Big\}$$

$$\text{s.t.} \quad \forall t = 1,\cdots,T, \forall \mathbf{c}_t \in \Omega_t:$$

$$\frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} y_{i_t}(\mathbf{w}_t\cdot\mathbf{x}_{i_t}) + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\cdot\operatorname{sign}(\mathbf{w}_t^{(k)}\cdot\mathbf{x}_{i_t})(\mathbf{w}_t\cdot\mathbf{x}_{i_t})\Big\} \geq \frac{1}{m_t}\Big\{C_l\sum_{i=1}^{l_t} c_{i_t} + C_u\sum_{i=l_t+1}^{m_t} c_{i_t}\Big\} - \xi_t. \tag{22}$$

The solution $(\mathbf{w}_t, \mathbf{v}_t, \xi_t)$ is then uploaded to the central server where the closed form of $z^{(k+1)}$, $u^{(k+1)}$ and the objective function value L can be derived as:

$$\begin{aligned}
\mathbf{w}_0^{(k+1)} =& \rho\sum_{t=1}^{T}(\mathbf{w}_t^{(k+1)} - \mathbf{v}_t^{(k+1)} + \mathbf{u}_t^{(k)})/(2+T\rho),\\
\mathbf{u}_t^{(k+1)} =& \mathbf{u}_t^{(k)} + (\mathbf{w}_t^{(k+1)} - \mathbf{w}_0^{(k+1)} - \mathbf{v}_t^{(k+1)}), \forall t = 1, \cdots, T,\\
L^{(k+1)} =& ||\mathbf{w}_0^{(k+1)}||^2 + \frac{\lambda}{T}\sum_{t=1}^{T}||\mathbf{v}_t^{(k+1)}||^2 + \sum_{t=1}^{T}\xi_t^{(k+1)}. \quad (23)
\end{aligned}$$

After the central server updates z and u, it scatters them back to each user to locally update x again until convergence. In this way, the individual users only need to communicate with the server and exchange the estimations of the parameters. There is no raw data involved in the communication and there is no information exchange between users. Thus the privacy of individual users are protected and the communication cost is also greatly reduced.

The ADMM loop can be set to stop when the norm of the dual residuals $\mathbf{s}^{(k+1)}$ and the primal residuals $\mathbf{r}^{(k+1)}$ are less than their thresholds $\sqrt{2T}\epsilon^{\text{abs}}$ and $\sqrt{T}\epsilon^{\text{abs}}$ respectively [56].

$$\begin{aligned}
||\mathbf{s}^{(k+1)}|| =& \rho\sqrt{2T}||\mathbf{w}_0^{(k+1)} - \mathbf{w}_0^{(k)}||,\\
||\mathbf{r}_t^{(k+1)}|| =& \sqrt{\sum_{t=1}^{T}||\mathbf{u}_t^{(k+1)} - \mathbf{u}_t^{(k)}||^2}. \quad (24)
\end{aligned}$$

The detailed steps are summarized in Algorithm 2.

Algorithm 2 : Distributed PLOS

```
1. Initialization. Set $\lambda, C_l, C_u, \epsilon, \mathbf{w}_0^{(0)}, \mathbf{u}_t^{(0)}$;
2. The CCCP process. The server applies first-order Taylor expansion to $|\mathbf{w}_t \cdot \mathbf{x}_{i_t}|$ to get the convex optimization (19);
3. The ADMM process. The server delivers $\mathbf{w}_0$, $\mathbf{u}_t$ to all users and each user set local $C_l, C_u, \epsilon^{\text{abs}}, \rho, \Omega_t = \phi$;
4. The cutting plane process. Each user solves the problem (22) to get $(\mathbf{w}_t^*, \mathbf{v}_t^*, \xi_t^*)$ and send them to the server;
5. The server calculates $\mathbf{w}_0, \mathbf{u}_t$, the objective function value $L$ using (23), and the residuals using (24);
6. If the residuals are less than the thresholds $\epsilon^{\text{dual}}$,$\epsilon^{\text{pri}}$, goto step 7; otherwise, goto step 3;
7. If $L$ does not converge, goto step 2.
```

VI. Experiments

In this section, we test the proposed method on both real-world and synthetic datasets. We also implement the distributed PLOS framework upon a real mobile sensing system. The experimental results show that PLOS performs considerably better than the state-of-the-art methods under a wide spectrum of scenarios. We first discuss the experiment setup in Section VI-A. We then present experimental results on the body sensor data in Section VI-B, on the smartphone data in Section VI-C, and on different scenarios of simulated datasets in Section VI-D. Finally, the distributed PLOS is tested on a smartphone platform in Section VI-E.

A. Experiment Setup

In this part, we introduce the baseline methods and the performance measures used in the evaluation.

Baseline Methods. We consider three types of baseline methods: a totally centralized method, a totally localized method and a group-based method. The details of the baseline methods are as follows.

- All. This baseline is a centralized method. All users are required to upload their data to the server along with the labels if there are any. The server will train a single global hyperplane from all the labeled samples, and apply this global hyperplane on the data of all the users.
- Single. This baseline is a localized method. Each user locally conducts classification/clustering based on only his own data. If a user has labels, then an SVM classifier is trained from the labeled samples. Otherwise, the k-means algorithm is applied to derive the clusters. The evaluation is also conducted locally. Since the cluster may mismatch with the ground truth labels, we conduct label matching on the clustering results and evaluate them under the best class assignments. On a specific type of users, we report the average.
- Group. This baseline is a group-based method. We measure the similarity between the users based on their sensory data. Specifically, given two users u, v, we first apply the random hyperplane algorithm [58] on their sensory data, which hashes the continuous sensory data to n discrete buckets while keeping the distance between the data. Let $(u_1, u_2, \cdots, u_n)$ and $(v_1, v_2, \cdots, v_n)$ represent the frequencies that the data of u and v appear in these buckets respectively, the similarity between u and v can be defined as the overlap between their sensory data, $S(u, v) = \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)}$, which is known as the Jaccard similarity coefficient. Knowing the similarity between the users, we further cluster similar users into a group through spectral clustering, within which the users share their data and labels. Finally, we conduct classification/clustering in each group and apply the learned hyperplane to all the users in that group. In all our experiments, n is set to be 128 and the number of clusters is set to be 3.

Performance Measures. In the experiments, to evaluate the performance, we adopt the accuracy of the classification/clustering results. More specifically, we apply the learned hyperplanes on the data and calculate the difference between the labels assigned by the hyperplanes and the ground truth labels. We report the accuracy on users with labels and without labels separately, since the methods may behave differently on different types of users. In addition, we select parameters for both the baseline methods and our proposed method based on the accuracy reported by leave-one-out cross-validation.

B. Experiments on Body Sensor Data

In this section, we build a body sensor network for each user to collect his/her motion information. In this sensor network, data are collected from multiple human motion sensing nodes placed in different body areas, and then we apply our personalized learning approach to verify its advantage.

Experimental Setup. The human motion sensing node in our experiment is TelosB, which carries a custom-built sensor board containing a triaxial accelerometer and a biaxial gyroscope. It also includes IEEE 802.15.4/ZigBee compliant RF transceiver so that data from all the nodes can be gathered at one base station through ZigBee.

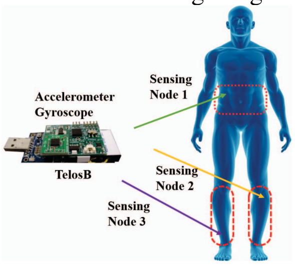


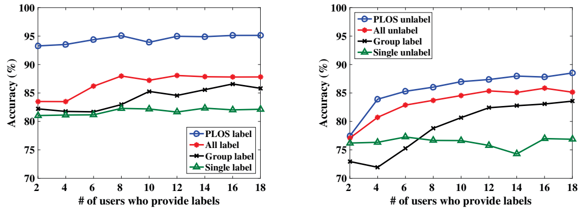
Figure 2. Sensors are placed on 3 different regions on the body: waist, left shin, and right shin.

20 subjects (age between 18 and 35) participated in our study. Each subject wore three sensing nodes on three different regions of his/her body, i.e., waist, left shin, and right shin as shown in Figure 2. In order to make the settings more practical, no instruction was given to the subjects regarding the exact placement and orientation of the sensing nodes and the subjects are allowed to place the devices anywhere in the requested body areas. They can also choose to attach the sensing nodes to the skin or to the clothes. Each subject wore the sensing nodes for 5 minutes, during which he/she was asked to perform two kinds of activities: rest at standing and rest at sitting.

The data collected from each sensing node contain 5 signals, i.e., x,y,z axis of the accelerometer and u,v axis of the gyroscope. They were first downsampled to 20 Hz and normalized. Then we split all the signals by a fixed-width sliding window of 3.2 seconds with 50% overlap, which generates 70 segments of accelerometer and gyroscope signals for each activity. Next, we convert each segment into feature vectors by extracting features from two aspects. The first aspect contains the features that can characterize each single signal, such as: mean, standard deviation, median absolute deviation (MAD), maximum, minimum, energy, interquartile range. The second aspect contains the features that are derived from several related signals: magnitude of three axes of accelerometer, the angles between the acceleration and the three axes, and signal magnitude area (the normalized integral of absolute value) of accelerometer output. Finally, the feature vectors of all the three sensing nodes are combined to create a feature vector of 120 dimensions.

Effect of the Number of Users Who Provide Labels. We first evaluate whether increasing the number of users who provide labels can help learn more accurate classifiers on the collected dataset. In order to do this, we gradually increase the number of users who provide labels, from 2 users to 18 users. Meanwhile, for users who provide labels, we set that they randomly labeled 6% of their data, (i.e., approximately 4 samples for each activity). We present the results in Figure 3.

From Figure 3, we can observe that the performance of Single on users with and without labels does not improve because the users do not have sufficient labeled data and they do not borrow information from their peers. All is able to learn a better global classifier when the number of people increases because they bring more labels, which overcomes the issue of insufficient labels in Single. But it still has the defect of ignoring the difference among different users. The performance of Group can also improve as more people provide label information. However, the improvement lags behind All because the increased label information cannot be used by the users in the other groups. PLOS further improves the performance since it can capture the structures of individual users' data and train a personalized classifier for each of them. The improvement is more obvious to users who provide labels, because the label information can strongly guide the discovery of the underlying data structures.

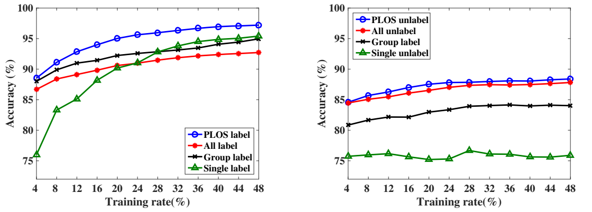


(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 3. Accuracy comparison on body sensor dataset w.r.t. the number of users who provide labels

Effect of the Size of Training Data. In this experiment, we randomly pick 9 users as label-providers and then observe how the number of labels they provide will influence their classification accuracy. In Figure 4, we vary the percentage of labels from 4% to 48% (i.e., around 3 samples to 34 samples out of the total 70 samples are labeled) and plot the classification accuracy of our approach and the two baselines.

All performs similarly compared with the previous experiment. In fact, the two experiments have an equivalent effect on All: the accuracy improves because there are more training data. Single performs poorly when the training data size is small, but improves dramatically on the users who provide labels as the training data increases and finally exceeds All. This implies that the users indeed demonstrate different patterns in their activities. However, the accuracy remains low on the users without labels. This is because that the users cannot help each other by sharing information. The performance of Group is in the middle. For users with labels, when there are enough labels, Group performs better than All but worse than Single because it only considers the individual difference among the groups, but not inside the groups. For users without labels, Group performs better than Single but worse than All because it only utilizes the labels inside the group but not those outside the group. By jointly modeling the commonness and differences among users, the proposed PLOS framework combines the strengths of All and Single, and conquers their weaknesses. Therefore, PLOS performs the best on all users in all scenarios.



(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 4. Accuracy comparison on body sensor dataset w.r.t. the percentage of labeled data on the users who provide labels

C. Experiments on Smartphone Data

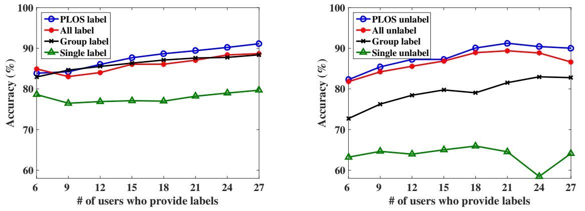
In this section, we conduct experiments on a real world mobile sensing dataset. The results clearly demonstrate the advantages of the proposed personalized learning framework.

HAR Dataset. UCI Human Activity Recognition (HAR) dataset [59] contains the recordings of 30 persons performing six different activities while wearing smartphones with embedded inertial sensors (accelerometer and gyroscope) on the waist. The activities include walking, walking upstairs, walking downstairs, sitting, standing, and laying. The readings form 561 features. In the following experiments, we consider the classification of sitting and standing, as this is the least separable pair among these six activities. There are around 50 samples for each activity from each person for this classification task, but only a very small portion (or none) of them are labeled.

To evaluate the performance of the proposed method, we conduct a series of experiments with different settings.

Effect of the Number of Users Who Provide Labels. Like previous experiment, we gradually add more users who provide labels, and we also set that they randomly label 6% of their data (i.e., around three samples for "sitting" and three samples for "standing" for each user who provides labels).

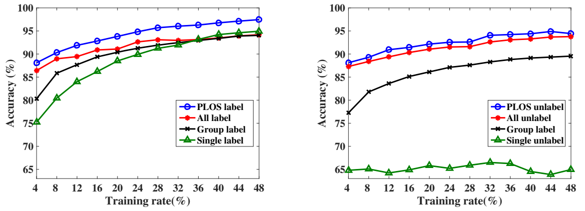
In Figure 5, we plot the accuracies on the users with labels and without labels with respect to the number of users who provide labels. It shows the same pattern as the experiment on body sensor dataset, except that gap of accuracy between All and PLOS is smaller. The reason may be that the body sensor dataset captures more personal traits of the users from two aspects. 1) sensor nodes placed on more body regions can provide a more complete view of the motion of a user; and 2) we allow the users to place the motion sensor anywhere in the required area, so users are more likely to place the motion sensors according to his/her personal habit, perhaps in different positions or different orientations. These reasons can also explain the phenomenon that Group performs similar to All and better than Single on users with labels. The results in this experiment further back up the advantage of the proposed PLOS framework.



(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 5. Accuracy comparison on HAR dataset w.r.t. the number of users who provide labels

Effect of the Size of Training Data. In this experiment, we randomly pick 15 users as the label providers. We gradually increase the number of labeled data in each of label provider' data. The results on labeled dataset are presented in Figure 6(a). The trend of all the methods is similar to that on the body sensor dataset. We find that when there are plenty of labels from users, the accuracy of Single and Group are closer to All than on the body sensor dataset. This conforms to our previous analysis that the body sensor dataset embodies more personal traits due to more sensing nodes and flexible experimental setup. In such condition, PLOS still performs the best among the three. The experimental results on unlabeled dataset are shown in Figure 6(b). Since the increase of the labeled data will not influence the clustering on unlabeled dataset, the accuracy of Single keeps low. When the number of training data increases, the performance of Group, All and PLOS increases. These patterns are similar to the experiment on body sensor dataset.

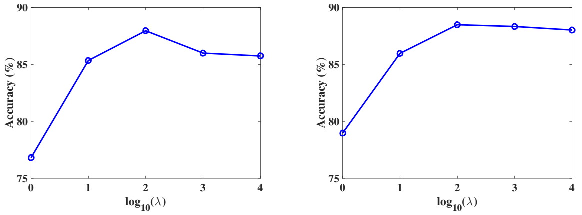


(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 6. Accuracy comparison on HAR dataset w.r.t. the percentage of labeled data on the users who provide labels

Effect of λ. As discussed earlier, the proposed PLOS framework performs the best because it can train personalized classifiers learned based on both individual data and the knowledge borrowed from other users. The balance of the two is controlled by λ. If λ is large, PLOS would enforce that the users share a similar hyperplane, so the results would lean towards All; If λ is small, then the results would lean towards Single. We examine the performance of PLOS with respect to $\log(\lambda)$ in the scenario that 15 users provide labels and they label 6 samples in their data. The results are presented in Figure 7.

It can be seen that the accuracy of all users reaches the best value when $\log(\lambda)$ is around 2. The performance degrades when $\log(\lambda)$ is either too small or too large. These experimental results confirm that there exist both commonness and difference in the activity patterns of different people.



(a) Accuracy on users with labels (b) Accuracy on users w/o labels

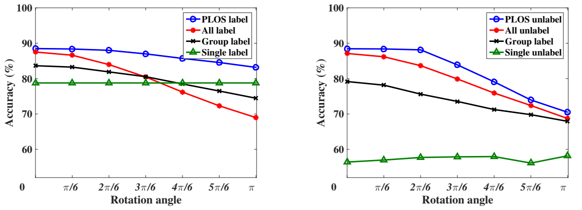
Figure 7. Accuracy comparison on HAR dataset w.r.t. the model parameters

D. Experiments on Synthetic Data

In this section, we use synthetic data to further test the proposed method under more general scenarios. In particular, we generate datasets containing two classes: $+1$ and -1. For each class, we generate 200 data points from a Normal distribution. More specifically, $Normal(\mu = (10, 10), \Sigma = \begin{bmatrix} 225 & -180 \\ -180 & 225 \end{bmatrix})$ for $+1$ class and $Normal(\mu = (-10, -10), \Sigma = \begin{bmatrix} 225 & -180 \\ -180 & 225 \end{bmatrix})$ for -1 class. To make the simulation more realistic, we randomly swap 10% of the ground truth labels, as in the real world applications, the data are rarely separable.

Effect of the Difference Levels among Users. In this experiment, we examine how the levels of differences among users can affect the proposed personalized leaning method. Intuitively, if the differences are large, then All may not perform well since it ignored the differences among the users. On the other hand, Single will not be affected much as it is trained on individual users' data. Group will be in the middle because it is able to learn different hyperplanes for different groups. The proposed PLOS method, considering both differences and commonness among the users, can perform much better than the baseline methods. To simulate different users, we first generate a data set from the aforementioned Gaussian distribution and then rotate the data around the origin with different angles. One rotation of the original data corresponds to the data from one user. Thus, with a given maximum rotation angle, we can generate 10 users with uniform rotation angles. Among the users, 5 of them provide labels for 8 samples (four from $+1$ class and four from -1 class).

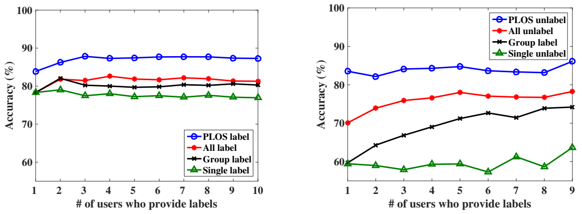
The experimental results are shown in Figure 8. The curves match our expectation perfectly. When the maximum rotation angle increases, the users are more different from each other, so the performance of All degrades quickly. The performance of Single stays the same and the performance of Group decreases slower than All, which are all expected. The accuracy of PLOS also decreases slightly but it is still the best. Note that the decrease is faster on the users without labels than the users with labels. This is because that when the users differ a lot, the users without labels cannot "borrow" too much helpful knowledge from the users with labels, and thus suffer more on the performance.



(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 8. Accuracy comparison on the synthetic dataset w.r.t. the rotation angles

Effect of Other Settings. Similar to the experiments on the real dataset, we also examine the performance of the proposed method with respect to the number of label providers and labeling percentage, respectively. In both experiments, we fix the maximum rotation angles to be $\pi/2$. For the former, we set the labeling percentage to be 2%, and for the latter, we set the number of label-providing users to be 5. The results are presented in Figures 9 and 10. The figures show similar patterns as the experiments on the real dataset, which confirms the advantages of the proposed method. In addition, in Figure 9 the standard deviation of PLOS decreases from 7.37% to 0.75% on users with labels and decreases from 8.39% to 3.19% on users without labels when the labeling percentage increases from 1% to 10%, which means as the labeling percentage increases, the uncertainty of the PLOS decreases on both types of users.

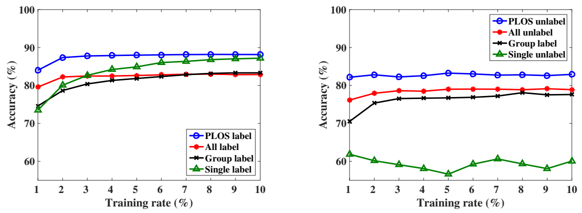


(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 9. Accuracy comparison on synthetic dataset w.r.t. the number of users who provide labels

E. Experiments on Distributed PLOS

In this section, we evaluate the accuracy and the scalability of the proposed distributed PLOS on a distributed mobile sensing system. The server is emulated by an Intel(R) Core(TM) 3.4GHz computer with 16GB of memory, and the users use Nexus 5 android phones as their sensing devices.

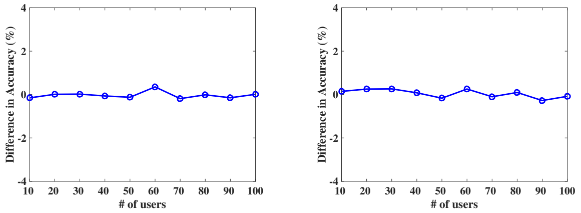


(a) Accuracy on users with labels (b) Accuracy on users w/o labels

Figure 10. Accuracy comparison on synthetic dataset w.r.t. the percentage of labeled data on the users who provide labels

Each user generates his/her own data as discussed before. In order to evaluate the performance of the distributed PLOS method on different scales, we vary the number of users from 10 to 100. We set the $\rho = 1, \epsilon^{\text{abs}} = 0.001$ as the step size and the stopping criteria respectively.

Accuracy Comparison with the Centralized PLOS. We first evaluate the classification accuracy of the distributed PLOS algorithm. As shown in Figure 11, the difference of accuracy between the distributed PLOS and the centralized PLOS is close to zero for both users with and without labels, which indicates that the distributed PLOS is a good approximation to the centralized PLOS.



(a) Difference of accuracy on users with labels (b) Difference of accuracy on users w/o labels

Figure 11. Difference of accuracy between the centralized PLOS and the distributed PLOS

Computational Cost. We further compare the running time of the centralized PLOS and the distributed PLOS. The centralized PLOS runs on the server and the running time is determined by how fast it can solve the optimization problem (4). On the other hand, the distributed PLOS allocates most of the calculation to smartphones. The smartphones conduct calculations in parallel, so the total running time is determined by the time consumption of the smartphone that processes the most amount of data. From Figure 12, we find that the centralized PLOS runs faster than the distributed PLOS when the number of users is small. However, as the number of users increases, the running time of centralized PLOS increases superlinearly, while that of the distributed PLOS almost keeps the same. This is because that adding more users increases the variables to the QP problem on the server, so it takes longer to solve the optimization problem. However, for the distributed PLOS, adding more users does not add variables to the QP problem on individual smartphones, so for each user, the running time stays almost the same.

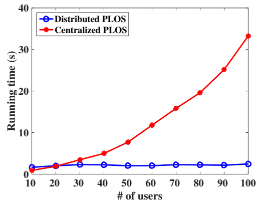


Figure 12. Running time for the centralized PLOS and the distributed PLOS

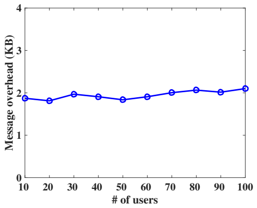


Figure 13. The message overhead of one user in the distributed PLOS

Communication Overhead and Convergence. Finally, we evaluate the number of messages that each user has to send to and receive from the server in the distributed PLOS. As discussed before, users do not upload their raw data to the server, but only exchange model parameters with the server. Thus, the communication overhead is determined by the total number of iterations that the algorithm needs to converge. We can see from Figure 13 that the message overhead of the individual users is reasonable and remains stable regardless of the number of users in the system. This result also implies that distributed PLOS converges at a stable rate.

VII. Conclusions and Future Work

In this paper, we study personalized learning in mobile sensing. The proposed PLOS framework enables personalized learning without requiring all of the users' data delivered to the server. It combines the knowledge from the users but at the same time also recognizes their uniqueness. The distributed nature of PLOS enables the knowledge sharing among the users with their privacy being preserved since the users neither upload the raw data nor communicate with other users. It is also efficient in terms of energy, computation, and communication costs.

In this paper, we mainly focus on SVM as it is one of the most widely adopted classification models. In the future, we will consider to extend the proposed framework to other machine learning models. Additionally, the current distributed algorithm is mainly designed for the synchronous distributed system. For the asynchronous scenario, for instance, some users may delay their responses for arbitrarily long, we will leave it as our future work.

Acknowledgment

This work was sponsored in part by US National Science Foundation under grant CNS-1652503 and CNS-1737590.

References

[1] P. Klasnja and W. Pratt, "Healthcare in the pocket: mapping the space of mobile-phone health interventions," *Journal of biomedical informatics*, vol. 45, no. 1, pp. 184–198, 2012.

[2] J. Ko, T. Gao, R. Rothman, and A. Terzis, "Wireless sensing systems in clinical environments: Improving the efficiency of the patient monitoring process," *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 2, pp. 103–109, 2010.

[3] T. Van Kasteren, G. Englebienne, and B. J. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *PerCom*, 2010.

[4] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, 2010.

[5] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *MobiSys*, 2012.

[6] A. Mehrotra, V. Pejovic, and M. Musolesi, "Sensocial: a middleware for integrating online social networks and mobile sensing data streams," in *Middleware*, 2014.

[7] X. Liu, J. Cao, S. Tang, Z. He, and J. Wen, "Drive now, text later: Nonintrusive texting-while-driving detection using smartphones," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 73–86, 2017.

[8] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *MobiCom*, 2009.

[9] S. Yang, F. Wu, S. Tang, T. Luo, X. Gao, L. Kong, and G. Chen, "Selecting most informative contributors with unknown costs for budgeted crowdsensing," in *IWQoS*, 2016.

[10] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao, "Enabling large-scale human activity inference on smartphones using community similarity networks (csn)," in *UbiComp*, 2011.

[11] X. Sun, H. Kashima, R. Tomioka, N. Ueda, and P. Li, "A new multi-task learning method for personalized activity recognition," in *ICDM*, 2011.

[12] X. Sun, H. Kashima, and N. Ueda, "Large-scale personalized human activity recognition using online multitask learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2551–2563, 2013.

[13] J.-H. Hong, J. Ramos, and A. K. Dey, "Toward personalized activity recognition systems with a semipopulation approach," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 101–112, 2016.

[14] J. W. Lockhart and G. M. Weiss, "The benefits of personalized smartphone-based activity recognition models," in *SDM*, 2014.

[15] F. Wu and Y. Huang, "Personalized microblog sentiment classification via multi-task learning," in *AAAI*, 2016.

[16] X. Miao, C.-T. Chu, L. Tang, Y. Zhou, J. Young, and A. Bhasin, "Distributed personalization," in *KDD*, 2015.

[17] T. Yu, Y. Zhuang, O. J. Mengshoel, and O. Yagan, "Hybridizing personal and impersonal machine learning models for activity recognition on mobile devices," in *MobiCASE*, 2016.

[18] S. Abdullah, N. D. Lane, and T. Choudhury, "Towards population scale activity recognition: A framework for handling data diversity." in *AAAI*, 2012.

[19] X. Bao, P. Bahl, A. Kansal, D. Chu, R. R. Choudhury, and A. Wolman, "Helping mobile apps bootstrap with fewer users," in *UbiComp*, 2012.

[20] N. D. Lane, Y. Xu, H. Lu, A. T. Campbell, T. Choudhury, and S. B. Eisenman, "Exploiting social networks for large-scale human behavior modeling," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 45–53, 2011.

[21] D. Peebles, H. Lu, N. D. Lane, T. Choudhury, and A. T. Campbell, "Community-guided learning: Exploiting mobile sensor users to model human behavior." in *AAAI*, 2010.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[23] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, 2013.

[24] D. Roggen, K. Förster, A. Calatroni, and G. Tröster, "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.

[25] A. Calatroni, D. Roggen, and G. Tröster, "Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion," in *INSS*, 2011.

[26] S. A. Rokni and H. Ghasemzadeh, "Plug-n-learn: Automatic learning of computational algorithms in human-centered internet-of-things applications," in *DAC*, 2016.

[27] ——, "Synchronous dynamic view learning: A framework for autonomous training of activity recognition models using wearable sensors," in *IPSN*, 2017.

[28] C. Persello and L. Bruzzone, "Active learning for domain adaptation in the supervised classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4468–4483, 2012.

[29] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised transfer component analysis for domain adaptation in remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3550–3564, 2015.

[30] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *IJCAI*, 2011.

[31] Y. Zhang and D. Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *UAI*, 2010.

[32] Y. Xue, D. Dunson, and L. Carin, "The matrix stick-breaking process for flexible multi-task learning," in *ICML*, 2007.

[33] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD*, 2004.

[34] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 615–637, 2005.

[35] Y. Ji and S. Sun, "Multitask multiclass support vector machines: model and experiments," *Pattern Recognition*, vol. 46, no. 3, pp. 914–924, 2013.

[36] J. B. Predd, S. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[37] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for big data systems," in *ICDCS*, 2015.

[38] Q. Jia, L. Guo, Z. Jin, and Y. Fang, "Privacy-preserving data classification and similarity evaluation for distributed systems," in *ICDCS*, 2016.

[39] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan, "Crowd-ml: A privacy-preserving learning framework for a crowd of smart devices," in *ICDCS*, 2015.

[40] K. Xu, H. Ding, L. Guo, and Y. Fang, "A secure collaborative machine learning framework based on data locality," in *GLOBECOM*, 2015.

[41] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *SP*, 2013.

[42] S. Isaacman, S. Ioannidis, A. Chaintreau, and M. Martonosi, "Distributed rating prediction in user generated content streams," in *RecSys*, 2011.

[43] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Transactions on Signal Processing*, vol. 63, no. 14, pp. 3700–3714, 2015.

[44] Q. Xu and R. Zheng, "When data acquisition meets data analytics: A distributed active learning framework for optimal budgeted mobile crowdsensing," in *INFOCOM*, 2017.

[45] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *MobiArch*, 2007.

[46] L. Ramaswamy, B. Gedik, and L. Liu, "A distributed approach to node clustering in decentralized peer-to-peer networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 9, pp. 814–829, 2005.

[47] P. Hu, S. S. Chow, and W. C. Lau, "Secure friend discovery via privacy-preserving and decentralized community detection," *arXiv preprint arXiv:1405.4951*, 2014.

[48] M. Halkidi and I. Koutsopoulos, "Online clustering of distributed streaming data using belief propagation techniques," in *MDM*, 2011.

[49] R. D. Nowak, "Distributed em algorithms for density estimation and clustering in sensor networks," *IEEE transactions on signal processing*, vol. 51, no. 8, pp. 2245–2253, 2003.

[50] H. Ding, L. Su, and J. Xu, "Towards distributed ensemble clustering for networked sensing systems: a novel geometric approach," in *MobiHoc*, 2016.

[51] J. Hamm, Y. Cao, and M. Belkin, "Learning privately from multiparty data," in *ICML*, 2016.

[52] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *NIPS*, 2005.

[53] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural computation*, vol. 15, no. 4, pp. 915–936, 2003.

[54] A. J. Smola, S. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *AISTATS*, 2005.

[55] J. E. Kelley, Jr, "The cutting-plane method for solving convex programs," *Journal of the society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[56] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[57] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[58] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *STOC*, 2002.

[59] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.