Hybridizing Personal and Impersonal Machine Learning Models for Activity Recognition on Mobile Devices

Tong Yu
Electrical and Computer Engineering
Carnegie Mellon University
tong.yu@sv.cmu.edu

Yong Zhuang
Electrical and Computer Engineering
Carnegie Mellon University
yongzhua@andrew.cmu.edu

Ole J. Mengshoel
Electrical and Computer Engineering
Carnegie Mellon University
ole.mengshoel@sv.cmu.edu

Osman Yağan
Electrical and Computer Engineering
Carnegie Mellon University
oyagan@andrew.cmu.edu

ABSTRACT

Recognition of human activities, using smart phones and wearable devices, has attracted much attention recently. The machine learning (ML) approach to human activity recognition can broadly be classified into two categories: training an ML model on (i) an impersonal dataset or (ii) a personal dataset. Previous research shows that models learned from personal datasets can provide better activity recognition accuracy compared to models trained on impersonal datasets. In this paper, we develop a hybrid incremental (`HI`) method with logistic regression models. This method uses incremental learning of logistic regression, to combine the advantages of both impersonal and personal approaches. We investigate two essential issues in this method, which are the selection of the learning rate schedule and the class imbalance problem. Our experiments show that the model learned using our `HI` method give better accuracy than the model learned from personal or impersonal data only. Besides, the techniques of adaptive learning rate and cost-sensitive learning give faster updates and more robust ML models in incremental learning. Our method also has potential benefits in the area of privacy preservation.

CCS Concepts

•**Human-centered computing** → **Empirical studies in ubiquitous and mobile computing;** •**Computing methodologies** → *Online learning settings;*

Keywords

Activity recognition, Incremental learning, Adaptive learning rate, Cost-sensitive, Privacy, Logistic Regression

.

1. INTRODUCTION

Human activity recognition based on sensor data has become very popular recently, partially due to the increasing availability of sensors in mobile devices and the advances in data analytics. Simply put, this approach is based on collecting data from various sensors, handcrafting a number of features, learning classifiers from the features, and then applying the learned classifiers to determine users' activities. Activity recognition has several applications, such as in healthcare [10, 9, 4] and mobile advertising [25].

Traditionally, there are two ways of training a machine learning model using data collected on mobile devices: on servers in the cloud [31, 26] and on the mobile devices [33, 22, 1]. However, it is often not sensible to train a general machine learning model on an impersonal dataset in servers and then move the model to different users' cell phones. The reason is that behaviors of different users may vary substantially and the collected data may be empirically sampled from different distributions. For example, the bus models and road situations vary in different countries. When users take buses in different countries, the data sensed by their smart phone acceleration sensor would be different. Therefore, a feature engineered based on this raw sensor data will probably be sampled from different distributions. As a result, it will be extremely difficult to predict user activities such as riding a bus.

To overcome this problem, for different user, we can specifically train a model in the cloud, based on that user's personal data. However, there are several potential problems also. First, we need to train different models for different users on servers and then send those models over the network, which can be problematic for server computational load and network bandwidth. Second, it may be difficult for servers to access enough personal data for all users. Another option would be to leave the training task to the mobile device. One shortcoming of this approach is that compared to cloud computing, the power and computational capability of a mobile device is quite limited. Besides, at the beginning when the devices start collecting users' personal data, the training data belonging to some activity classes could be very limited, which in turn would make the trained model's generalization ability for those classes quite poor.

In this work, we propose and carefully evaluate a hybrid incremental (HI) method to effectively address, from two angles, the problems mentioned above. First, at the beginning of the process we train an impersonal machine learning model in the cloud using an impersonal dataset. Then we send this model to the mobile devices of different users. After receiving the impersonal model, each device will then decide whether it should further train the model incrementally; this decision will be based on whether the device has (enough) personal and labeled data. We study the logistic regression learning model in this paper, considering its extremely small model size that saves bandwidth, good performance in activity recognition, and easy incremental update.

Unfortunately, the logistic regression model faces two problems in this particular incremental learning task. First, with the variations in the behavior of different users, it is often difficult to select a universally suitable learning rate across all users. Second, the personal data is usually class imbalanced. Usually, it is easier to collect the data of certain activities such as being still, walking, and running, compared to other activities including biking and taking the bus.

To avoid jeopardizing the logistic regression model after incremental learning, we carefully examine the adaptive learning rate and apply cost sensitive techniques to address these two problems.

The contributions of this paper are as follows.

- We propose and thoroughly evaluate an HI method for an activity recognition system that combines the benefits of training the model on universal impersonal data and on personal data by incremental learning.
- We point out a non-trivial problem in our method, namely that of selecting the learning rate. We adopt an adaptive learning rate technique to deal with this problem.
- We show that there is a potential class imbalance problem on personal datasets in practice, and address it using cost sensitive machine learning techniques.

The rest of the paper is structured as follows. Section 2 discusses related work. In Section 3, we present our hybrid incremental (HI) method for activity recognition. It combines the advantages of two traditional methods: training a machine learning model on an impersonal dataset and on a personal dataset. In Section 4, we discuss our method for tuning the learning rate and resolving the class imbalance problem. Section 5 is devoted to empirical results that validate our analysis and justify the methods we developed. We conclude the paper in Section 6 with discussion of future research directions.

2. RELATED WORK

2.1 Activity Recognition

With high availability of sensors in mobile devices becoming the norm, and data analysis and machine learning techniques getting more and more advanced, human activity recognition is becoming a popular and solvable problem.

A typical activity recognition system [31, 26, 16, 22, 1, 2, 19, 24] works as follows: Data is collected from various sensors (*e.g.*, accelerometer, magnetometer, GPS, gyroscope) and stored in a log of raw data. From this raw data, meaningful features, such as mean, variance, and FFT in a fixed size window, are generated through *feature engineering* techniques. Then a machine learning model is trained on the engineered features. The obtained machine learning model is then used to predict a user's activity for specific feature values. With this activity recognition engine, users' daily activity and health can be monitored. For example, we can calculate users' daily calorie consumption and encourage users to do more exercise when necessary.

In general, based on the training process, this research can be classified into two categories: training a machine learning model on impersonal datasets on the *server* [31, 26, 21], or on the *mobile device* [21, 33, 22, 1]. Lockhart and Weiss [21] created a comprehensive comparison between training a model using an impersonal dataset and a personal dataset. This study concluded that with careful control of the setting, training on a personal dataset can effectively improve the recognition accuracy. Another previous work [17] combine data from multiple users and devices based on their similarities, to achieve better prediction performance.

2.2 Incremental Learning on Device

Incremental learning is an important technique in machine learning. Usually, we train a model on a dataset and use it for prediction. When more and more new data are collected, we need to update the model. If we train the model from scratch every time, the costs would be very high. Incremental learning incorporates the old model with the newly available data and avoids re-training from scratch.

Several previous works focus on different types of data in incremental learning problems. For example, Bifet and Gavalda [6] focused on time-varying data via an adaptive windowing approach. Gao et al. [12] handled concept-drifting data streams with skewed distributions. There are some works specifically on incremental learning in activity recognition. For instance, Longstaff et al. [22] proposed incremental learning and adaptation in streaming settings. Active and semi-supervised learning was applied to improve activity recognition [1]. Zeng et al. [33] proposed a dynamic heterogeneous sensor fusion framework, which incrementally updates the weights of different sensors.

In this paper, we aim to develop a robust incremental learning technique for logistic regression model in activity recognition. Logistic regression is chosen here for several reasons. First and foremost, its small model size fits well to our proposed method. Also, it is scalable, enables parallel training in servers and fast incremental training in devices, and has high recognition accuracy; see Section 3.2 for details.

2.3 Existing Learning Rate Schedules

The stochastic gradient method (SG) is an effective optimization method in machine learning [34, 15, 7]. However, SG's performance is highly sensitive to the selection of learning rate. Previous works have focused on addressing this problem [11, 32, 8], which can be divided into three types according to the way they set the learning rate:

- **Fixed Learning Rate**: During training, the learning rate is fixed to a pre-specified constant.
- **Adaptive Learning Rate**: Based on the objective function value, the learning rate is adjusted dynamically during training (e.g, [13]).
- **Per-coordinate Adaptive Learning Rate**: Different from the previous two types of learning rate sched-

ules, it applies different learning rates to each coordinate of the model (e.g., [11, 27, 32]).

Chin et. al. [8] have demonstrated the effectiveness of per-coordinate adaptive learning rate among the above schedules. Taking computational simplicity and cost into consideration, we choose the adaptive gradient algorithm (ADAGRAD) [11] in our problem as our adaptive learning rate schedule.

2.4 Class Imbalance

We regard human activity recognition as a multi-class classification problem. In the real world, users' data in different activity categories are not evenly distributed. For example, we have more data of walking, compared to that of jogging. With these highly-skewed class distributions, a model may tend to output the majority class as its prediction. To reduce the influence resulting from class imbalance, there are two strategies: sampling (*e.g.*, up-sampling and down-sampling) and cost-sensitive learning.

Seiffert et al. [28] presented a comparative study of data sampling and cost sensitive learning. They showed that down-sampling and cost-sensitive learning outperformed other techniques. Down-sampling method is based on removing some of the data to change the distribution. However, some users' labeled data can be very limited for some activities. Therefore, we will use cost-sensitive strategy instead of the down-sampling technique.

3. HYBRID INCREMENTAL METHOD

First, our proposed hybrid incremental (`HI`) machine learning method is introduced. Then, it is explained why logistic regression is selected as the machine learning model in our proposed method. We further discuss the some advantages and disadvantages of this method.

3.1 An Overview of the HI Method

Figure 1a shows a traditional way of training an ML model for an activity recognition system. First, the server collects all available data and trains a model. If the dataset is too big for a single machine, a distributed ML platform (*e.g.*, Hadoop, Spark) can be applied. Second, the server sends the ML model to the mobile devices. Users will get the prediction results from a mobile app, based on this ML model. A closely related approach is that the app uploads the collected sensor data to the server, and the server returns the prediction results of the ML model.

Figure 1b shows, in contrast, our proposed hybrid incremental method (`HI`). First, a server collects all available data and trains an ML model. Second, the server sends the ML model to the mobile devices. In this `HI` method, we have a additional third step for incremental updating of the model on the mobile devices based on the collected personal data.

The following categories define our above approach as well as several other approaches discussed in Section 2.1:

- **Impersonal:** The models are trained on a general dataset in remote servers directly. This is a very common approach, which is shown in Figure 1a.

- **Personal:** The models are trained on one user's dataset, and used to predict this particular user's activity.

Method	**Training**		**Test**
	Server	**Device**	
Impersonal [31, 26, 21]	D_s	$\emptyset$	M_s, D_j^{te}
Personal [21]	$\emptyset$	D_j^{tr}	M_j, D_j^{te}
Hybrid [21]	D_s	D_j^{tr}	M_{s+j}, D_j^{te}
Hybrid Incremental (ours)	D_s	M_s, D_j^{tr}	M_{HI}, D_j^{te}

Table 1: A comparison between Impersonal, Personal, Hybrid and `HI` Methods. D_s is the impersonal data on the server which includes all users, except the users in the test set. For user j, D_j^{tr} is labeled data that is collected the user's device while D_j^{te} is data for test. M_s is trained on D_s, M_j is trained on D_j^{tr}, M_{s+j} is trained $D_s \cup D_j^{tr}$ and M_{HI} is incrementally trained on D_j^{tr} with M_s as the initialization. If there is no model or data involved, it is $\emptyset$.

- **Hybrid:** The user's data is combined with a general dataset in servers. A model is induced from this combined dataset.

- **Hybrid Incremental (`HI`):** First, we train a model on a general dataset in remote servers. Then, the model is sent to the user devices. Finally, this model is incrementally updated based on collected data from a specific user. This is our proposed approach.

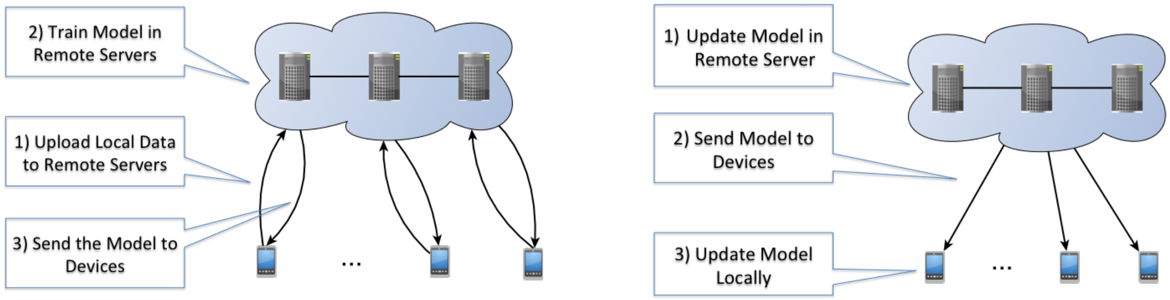
  Compared to **Hybrid** method, the training dataset of `HI` method is the same. However, our approach gives more weights to personal data during training models and provides better prediction performance, which is validated in Section 5.4. This method is demonstrated in Figure 1b.

In detail, we split the dataset user-wise into two groups D_s and D_d. D_s is the data stored in remote servers, while D_d is the data we collected in users' devices. Each user either belongs to D_s or D_d, because we assume that the actual users do not appear in the training data. That is, $D_s = \{D_1, D_2, \cdots, D_n\}$, where n is the number of users in the training data. We also have $D_d = \{D_{n+1}, D_{n+2}, \cdots, D_{n+m}\}$, where m is the number of users in the test data. For each user j in D_d, where $n+1 \leq j \leq n+m$, we split his or her data D_j into two parts, D_j^{tr} and D_j^{te}.

The difference between the Impersonal, Personal, Hybrid and `HI` methods is summarized in Table 1. In the Impersonal approach, for a particular user j, we train a model M_s on impersonal data D_s, and make prediction on D_j^{te}. In the Personal approach, we train a model M_j on personal data D_j^{tr}, and make prediction on D_j^{te}. In the Hybrid approach, we train a model M_{i+j} on the combined dataset $D_s \cup D_j^{tr}$. Then, model M_{i+j} is tested on data D_j^{te}. In our method, we train a model M_s on impersonal data D_s. Then, the model M_s is sent to different users. For each user j, M_s is used as the initial model for incremental updating. The data used in incremental learning is D_j^{tr}. After incremental learning we have an updated model M_{HI}, which is also tested on data D_j^{te}.

3.2 Logistic Regression as Classifier

Multiple models can be considered in our incremental hybrid method, such as decision tree, AdaBoost, support vector machines, random forest, etc. We will focus on logistic regression in this paper. In this case, the model mentioned



(a) Traditional server-centric method. It trains ML models based on an impersonal dataset on a server, and then sends the learned model to mobile devices. There are no model updates on the devices.

(b) Proposed HI method. It trains ML models based on impersonal dataset in the server, and then sends the model to the mobile devices. After obtaining the model, a mobile device can incrementally update the model.

Figure 1: A comparison between the traditional server-centric method and the proposed HI method.

in Section 3.1 is a weight vector $\mathbf{w}$. The model $\mathbf{w}$ is initialized as M_s of the HI method in Table 1 before incremental update. After incremental learning is finished, M_{HI} is the updated $\mathbf{w}$. If training instances are $\mathbf{x}_i, i = 1, \ldots, l$, and labels are $y_i \in \{1, -1\}$, logistic regression aims to optimize the following loss function:

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}),$$

where $C > 0$ is a parameter used to keep the two terms balanced, $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ is a regularization term, and l is the number of training instances. Logistic regression handles binary classification problems. In order to make predictions on multi-class datasets, we apply the one-against-all method [14]. To handle a dataset with k classes, this method constructs k binary models. Each of the k models is trained by treating one class as positive and all the remaining classes as negative.

The reasons why we choose logistic regression as the machine learning model in our HI method are listed below.

- Compared with other approaches such as decision tree, AdaBoost and random forest, logistic regression's model[1] size can be extremely small, even after using explicit kernel feature mapping [31]. Besides, when the size of training data increases, the model size of decision tree, AdaBoost and random forest will typically increase as well. However, for logistic regression the model size will stay the same, and grow linearly with the dimensionality of the data. Therefore, logistic regression typically will use less bandwidth to send the model and require less space in devices, compared to its alternatives.
- Logistic regression has provided competitive results in previous activity recognition tasks [31], and typically achieves very similar results to support vector machines. However, its logistic loss can provide better probability interpretations for activity recognition. The sigmoid function scales the decision values in to the range of $[0, 1]$. If a user is classified as walking, we can easily predict the probability of walking by calculating the logistic loss. Furthermore, with the explicit kernel mapping method, we can easily utilize high-dimensional data while still enjoying small model size.
- The loss function of logistic regression model is differentiable and the model can easily be trained by the stochastic gradient (SG) method. We can also easily design incremental SG methods to update a logistic regression model. On the server side, we can still use complicated but advanced optimization solvers for logistic regression model training, such as trust region Newton method [20] and even its scalable distributed version [35]. Thus, we can easily train new models on big activity data.

[1]Intuitively, logistic regression model is a type of support vector machines with logistic loss function instead of hinge loss. But logistic loss can provide better probability interpretations of predictions, which is very useful in activity recognition.

3.3 Discussion of the Method

In this section, we discuss the potential advantages and disadvantages of this method.

3.3.1 Potential Advantages

Our novel incremental hybrid method enjoys several advantages:

- **Low Bandwidth Consumption:** In the HI method, when users have new data, they do not upload their data to remote severs to retrain a model. Moreover, the servers do not send models to different users when the model is updated. Thus, bandwidth can be saved in practice. With the newly collected data, the model is updated in each user's device. Further, the logistic regression model requires very little storage space [31].
- **User Privacy:** In the HI method, each user does not need to upload their personal data to a central server, so their personal data and privacy are potentially better protected.
- **Efficient Update of Model:** When a user creates new data, the device can incrementally update the ML

model in a real-time fashion. There is no need to upload the new data and wait for the server to finish the training and send back the updated model.

3.3.2 Potential Disadvantages

Although our `HI` method enjoys several advantages, in practice it also faces two potential yet very important problems.

- **Learning Rate Selection** [8]: Usually, the behaviors of different users are quite different. Thus the empirical distributions of different users' activity data vary substantially, especially when the amount of personal activity data is extremely small. Thus, the optimal learning rate in SG for different users can vary significantly. Optimizing the learning for a particular user is a non-trivial problem.
- **Class Imbalance**: People tend to collect far more data of being still and walking, than data of being running or driving. Unfortunately, some machine learning algorithms, such as support vector machines or logistic regression, prefer to predict the majority class [29, 3, 30]. Their performance will degrade when the data is extremely imbalanced.

4. MACHINE LEARNING METHODS IN HI

In this section, we discuss our hybrid and incremental learning approach, and how to leverage different machine learning techniques to handle the potential issues faced by this method. First, we train a machine learning model from scratch on impersonal data in the servers. We discuss a few reasons to focus on logistic regression as the classifier in our method. We can use off-the-shelf learning methods such as SG or the Newton method to train this model. After getting the tuned model, we send it to the mobile devices for incremental training.

4.1 Incremental Learning On Devices

Incremental learning is a machine learning technique that takes an existing model and adjust it based on new examples. Thus, we can apply incremental learning to update the model without re-training from scratch.

In our method, each device downloads a logistic regression model $M_{HI} = \mathbf{w}$ from the remote server as the initial model. Then, based on the model $\mathbf{w}$ from the remote sever, we update it using new collected data from the device. Our incremental learning workflow is summarized in Algorithm 1.

4.2 Adaptive Learning Rate

We apply the Stochastic Gradient (SG) method for incremental learning [18]. SG is sensitive to the selection of learning rate, and parameters tuning can be a slow and toilsome process [11, 32, 27]. As introduced in Section 2.3, we apply one of the per-coordinate adaptive learning rate algorithms, ADAGRAD, to address this problem.

ADAGRAD normalizes the gradients in the current iteration via the past gradients. Then, ADAGRAD use the square root of the sum of gradients to normalize the learning rate for each selected sample i:

$$\eta_{i,k} = \frac{\alpha}{\beta + \sqrt{\sum_{j=0}^{i} \nabla f(w)_{j,k}^2}} \text{ for } k = 1, \dots, n, \qquad (1)$$

Algorithm 1: Incremental learning by ADAGRAD for logistic regression model.

Data: Training data on device (mobile phone) $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_l, y_l)$.
Input : Model $\mathbf{w}$ from the remote server, regularization parameter C and number of iteration N_{iter}, α, β, and $\sigma_k = 0$ which stores the sum of gradients for each dimension k.
Result: Updated model $\mathbf{w}$.
for $m \to N_{iter}$ **do**
  for $i \to l$ **do**
    $\nabla f(\mathbf{w}) = \mathbf{w} + C \sum_{i=1}^{l} (1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})^{-1} y_i \mathbf{x}_i$
    $\sigma_k = \sigma_k + \nabla f(\mathbf{w})_k^2$ for $k = 1, \dots, n$
    $\eta_k = \frac{\alpha}{\beta + \sqrt{\sigma_k}}$ for $k = 1, \dots, n$
    $\mathbf{w}_k = \mathbf{w}_k + \eta_k \nabla f(w)_{i,k}$ for $k = 1, \dots, n$

where α and β are user-specific parameters and fixed during training process, and $\nabla f(w)_{i,k}$ is the k^{th} dimension of the gradient of each selected instance i.

4.3 Cost-Sensitive Logistic Regression

We observe later in Section 5.7 that the classes are imbalanced. Thus, we use a simple yet easy method to implement a cost-sensitive strategy to solve this problem. In contrast to traditional logistic regression, it optimizes the following loss function.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ & + C^+ \sum_{i=1}^{l} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) \delta(y_i > 0) \\ & + C^- \sum_{i=1}^{l} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) \delta(y_i \le 0) \qquad (2) \end{aligned}$$

where C^+ and C^- mean the cost for the positive class and negative class respectively, and $\delta(x)$ is an indicator function:

$$\delta(x) = \begin{cases} 1, & \text{if statement } x \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

We set C^+ as the number of positive instances while C^- is the number of negative instances.

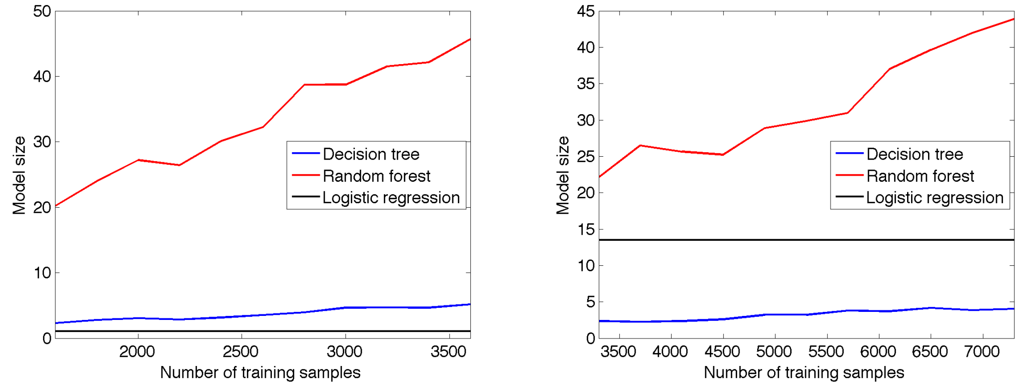
5. EXPERIMENTAL RESULTS

In this section, we conduct several experiments to demonstrate the validity of the proposed methods.

5.1 Dataset

We use the data in `Activity Prediction Data`[2] [16] and Human Activity Recognition Using Smartphones Data Set in `UCI Data`[3] [5]. The two datasets contain data from different volunteers performing different activities (*e.g.*, walking, jogging, etc.). Some of the statistics are listed in Table 2.

[2]http://www.cis.fordham.edu/wisdm/dataset.php
[3]https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones, which we refer as `UCI Data` in the rest of this paper.



(a) On `Activity Prediction Data`.

(b) On `UCI Data`.

Figure 2: How the model size changes when there more training samples. The x-axis shows the number of training samples and the y-axis shows the model size. For decision tree and random forest, the model size increases approximately linearly with the training sample size. For logistic regression, the model size is constant.

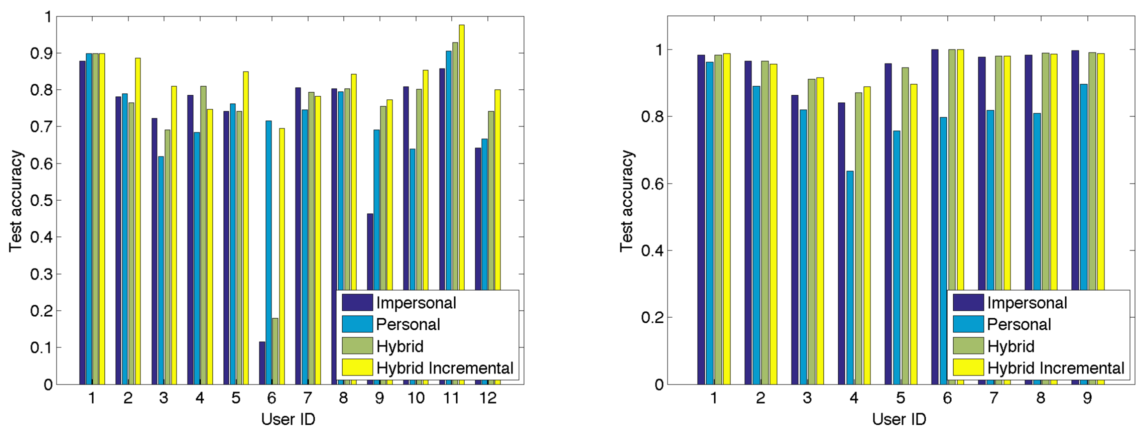


Figure 3: A comparison between Impersonal method, Personal method, Hybrid method and `HI` method. The x-axis shows the ID of different test users and the y-axis shows the test accuracy. The left panel shows the results on `Activity Prediction Data` and the right panel shows the results on `UCI Data`. On `Activity Prediction Data`, the incremental model performs best on 8 among 12 users. On `UCI Data`, the incremental model performs either best or very similar to the impersonal or hybrid methods.

Data	**# Users**	**# Activities**
`Activity Prediction Data` [16]	36	10
`UCI Data` [5]	30	6

Table 2: Number of users and number of activities in two evaluated datasets.

	Training		**Test**
	Server	**Device**	
(1)	21 users	about 25% of 9 users	about 75% of 9 users
(2)	24 users	25% of 12 users	75% of 12 users

Table 3: A table indicates how training and test data is split for dataset (1) `UCI Data` and (2) `Activity Prediction Data`. Here, 25% of 9 users indicates that for each user among the 9 users, 25% data are used for incremental training and the remaining 75% is used for testing.

5.2 Experimental Setting

To simulate the setting discussed in Section 3.1, for Human Activity Recognition Using Smartphones Data Set in UCI Data [5], we split 30 users into two groups with 21 users and 9 users respectively. This follows the original split into training and test sets. We place 21 users' data on the server and 9 users' data are on their separated devices. For each one among these 9 users, we assume he or she has 10 samples known[4] in each class of activity for incremental training and report the test accuracy on the rest of the samples. For `Activity Prediction Data` [16], we split 36 users into two groups with 24 users and 12 users respectively. 24 users' data are placed on the server and 12 users' data are on their own devices. For each one among the 12

[4]For each user in this dataset, 10 data is about 25% of all his data in each class of activity.

Machine Learning Model	Test Accuracy	Model Size
Decision Tree	0.7174	5.264 KB
Random Forest (2 trees)	0.7262	17.984 KB
Random Forest (5 trees)	0.7749	46.864 KB
Support Vector Machines	0.7397	1.080 KB
Logistic Regression	0.7398	1.080 KB

Table 4: Test accuracy and model size comparison for different learning models, on `Activity Prediction Data`.

Machine Learning Model	Test Accuracy	Model Size
Decision Tree	0.8798	4.272 KB
Random Forest (2 trees)	0.8340	15.840 KB
Random Forest (5 trees)	0.9048	45.104 KB
Support Vector Machines	0.9580	13.464 KB
Logistic Regression	0.9581	13.464 KB

Table 5: Test accuracy and model size comparison for different learning models, on `UCI Data`.

users with devices, we assume he or she has 25% data known in each class of activity for incremental training and report the test accuracy on the remaining 75% of data. The reason we split this data in this way is that in the transformed dataset `Activity Prediction Data` provided, the data is not well distributed. That is, for some users in some classes of activities, the data contains fewer than 10 samples.

The details about how to split the dataset is listed in Table 3.

5.3 Model Size Experiments

In this section, we compare different machine learning model sizes. We also present corresponding results for their test accuracy. We note that a model with *small* model size and competitive accuracy is suitable for our HI Method.

In the comparison, we use the *decision tree*, *random forest* and *support vector machines* as baseline approaches. Table 4 and 5 show the results on `Activity Prediction Data` and `UCI Data` respectively, from which several observations can be made. First, compared to decision tree and random forest, logistic regression usually achieves competitive accuracy and smaller model size. Random forest has high accuracy but large model size, while decision tree with small model size typically can not predict as well as other models. Second, logistic regression models and support vector machines usually achieve very similar performance while also having the same model size. As mentioned in Section 3.2, one advantage of logistic regression is that it has a clear probability interpretation for the predictions.

Another nice property of logistic regression compared to decision tree and random forest is that its model size is constant irrespective of the size of the data used in training the model. As shown in Figure 2, when we keep increasing the amount training data, the model size of decision tree and random forest will increase linearly. However, the model size of logistic regression will not change.

In real world, the training data stored in a server is expected to be much bigger than the data we use in experiments. Besides, the server data will keep increasing when more labeled data is collected. Thus, it is concluded that the logistic regression with constant model size is a better choice.

In view of the experimental results and previous discussion in Section 3.2, the rest of experiments will focus on the evaluation of logistic regression model.

5.4 Model Accuracy Experiments

Models trained from hybrid datasets can lead to better performance than models trained from impersonal datasets [21]. And the model trained from personal dataset performs the best among the three methods. The purpose of this section is to test the `HI` method against the other methods in Table 1.

In the experiments, we split the data as described in Section 5.2 and report the results on both two datasets. The experimental results are shown in Figure 3. It can be observed that the `HI` method performs the best among the four methods in most cases. Another observation is that the incremental learning's improvement are most significant in the cases when the accuracy of model obtained from server is not that good. In this case, those badly predicted users' behavior should be very different from those users whose data are placed on the server. But as incremental learning on personal data in the device can capture information on personal data, it can provided better personal prediction. The improvement by `HI` on `Activity Prediction Data` is smaller than `UCI Data`. The reason is that `UCI Data` is easier than `Activity Prediction Data`, with higher average accuracy. It means that the different users from `UCI Data` have less diversity. As a result, incremental learning on personal datasets do not help too much in improving test accuracy. In the experiments in following sections, we will focus on the relatively difficult `Activity Prediction Data`.

`HI` method outperforms personal and impersonal methods. The improvement can be explained by making use of the additional information on the each user's device. In the real world setting, if users are not that concerned about their privacy and willing to upload their personal data to the servers, we can also train a new model based on a combined dataset with both impersonal information and personal information, which we refer as traditional hybrid approach.

The incremental learning in `HI` method also provides better results than retraining a new model on servers in traditional hybrid approach. It can be interpreted by the process when we optimize the loss function of logistic regression model. When we retrain a model based on impersonal data and uploaded personal data in servers, we regard every single data point as equal. That is, every data point has the same weights when we calculate the loss function. However, in the incremental learning model in `HI` method, the weight of personal data have higher weights than impersonal data during the optimization process. After we train an impersonal model in servers, it can be expected that the obtained model is quite close to the a general optimal solution. In the incremental learning, we further refine this obtained the model according to the loss function calculated only by one user's personal data. In this case, the model is well adjusted to provide a more personal optimal solution for this particular user. This explains why incremental learning in `HI` method outperforms traditional hybrid approach.

User with ID of 6 on `Activity Prediction Data` in Figure 3 is an interesting example. The original impersonal data in servers may be too different from this user's data, so that model trained on impersonal data provides bad predictions. However, if we train on personal data, the model can predict very well. Simply combining the impersonal and personal data, and training a model won't help too much, as

Test User ID	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}
Learning rate 10^{-3}	0.878	0.78	0.722	***0.785***	0.741	0.126	***0.806***	0.803	0.473	0.809	0.857	0.642
Learning rate 5×10^{-4}	0.878	0.772	0.73	0.696	0.755	0.0632	0.764	0.803	0.773	0.809	***0.988***	0.675
Learning rate 10^{-4}	0.878	0.772	0.754	0.696	0.748	0.505	0.758	0.803	0.773	0.801	***0.988***	0.658
Learning rate 5×10^{-5}	0.878	0.829	0.794	0.696	0.82	0.632	0.782	0.803	0.773	0.824	***0.988***	0.692
Learning rate 10^{-5}	0.878	0.87	0.802	0.696	0.827	***0.695***	0.77	0.819	0.773	0.846	***0.988***	0.733
Adaptive learning rate	***0.898***	***0.886***	***0.81***	0.747	***0.849***	***0.695***	0.782	***0.843***	***0.773***	***0.853***	0.976	***0.8***

Table 6: Test accuracy with different learning rates and adaptive learning rate, on `Activity Prediction Data`. Columns $U_1, U_2, \cdots, U_{12}$ show different IDs of randomly selected test users. The italics font highlights the best ones for each user and adaptive learning rate performs best on 9 among 12 users. The underline marks the best ones of fixed learning rates. We can observe that the optimal fixed learning rates for different users vary a lot.

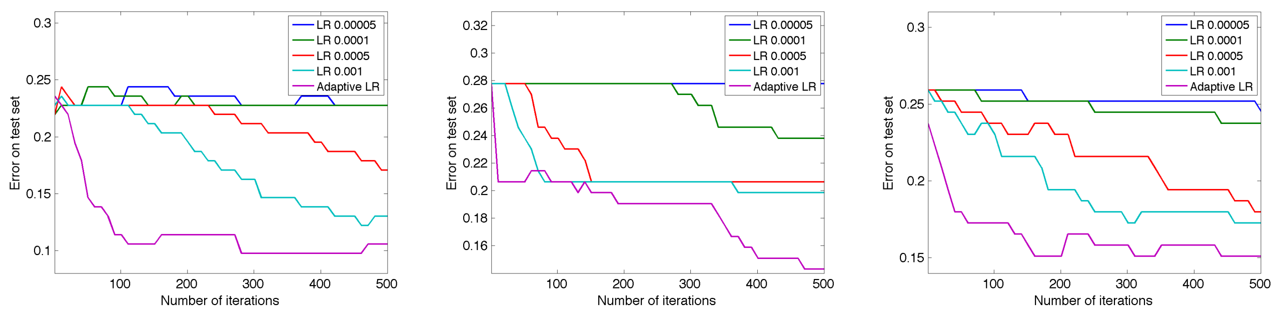


Figure 4: A comparison of the learning curves between several fixed learning rates (abbreviated as LR) and adaptive learning rate. The x-axis shows the number of iterations of learning and the y-axis shows the test accuracy. From `Activity Prediction Data`, three randomly selected users' results are presented.

hybrid's result shows. In `HI` method, we give higher weight to personal data during model training. Although hybrid approach uses exact the same data with our approach, we can provide model with much better performance.

5.5 Accuracy Comparisons between Different Learning Rates

In the real world, the incremental learning dataset is very small and the users' behavior diverse. The optimal learning rate for different users' data should be different as well. How to tune the most suitable learning rate for all users is very tricky. Adaptive learning rate may be a solution and we compare it with several fixed learning rates. We follow previous works [23, 8], and set $\alpha = 0.1$ and $\beta = 1$ in the following experiments.

In Table 6, the optimal fixed learning rate for different users varies, as marked by the underline. However, we can also observe that the adaptive learning rate can always achieve comparable performance with optimal learning rate. Thus, in real world application, it can helpfully select the most suitable learning rate automatically.

Similar with the observation in previous work [8], in our experiments the adaptive learning rate not only determines the suitable learning rate automatically, but also leads to faster convergence. This is demonstrated in Figure 4, where we observe that the adaptive learning rate's curves are significantly lower than those of fixed learning rates. Although the learning rate can be reduced significantly by using a larger fixed learning rate, this would also lead to inaccurate solutions when the optimization is close to the optimum, which is shown in Table 6. These results show that, in the incremental learning in `HI` method for the activity recognition problem, using adaptive learning can lead to both improved recognition results and reduced training effort in users' devices.

5.6 Convergence Speed Experiments

In addition to higher accuracy, another benefit of the `HI` method is that it can provide better convergence and less training time, compared with personal method on the same level of training personal data. For fair comparison, we use both adaptive learning rate for two methods. Figure 5 shows the convergence of the learning algorithm on some users' personal data. The error on test data in each iteration are compared between personal method and `HI` method. The incremental models in `HI` method have lower test error after convergence. Besides, it can be observed that the curve of incremental model in `HI` drops faster than the curve with personal method.

In practice, it means that with the information from a general model trained on impersonal data in server can not only help improve the activity recognition accuracy on personal dataset, but it can also help reduce the training time on user's device.

5.7 Class Imbalance Experiments

This section investigates cost-sensitive logistic regression's performance in the class imbalance problem. Class imbalance is very common in activity recognition. The class distribution in our experiments is presented in Figure 6.

In Figure 7, we show the difference between cost-sensitive logistic regression and cost-insensitive logistic regression. In these two approaches, we use adaptive learning rate to incrementally update the model. It can be observed that in most cases, the cost-sensitive setting can achieve the best performance. For the user with ID 6, the cost-sensitive approach is not helpful. In this case, we observe that there is no up-

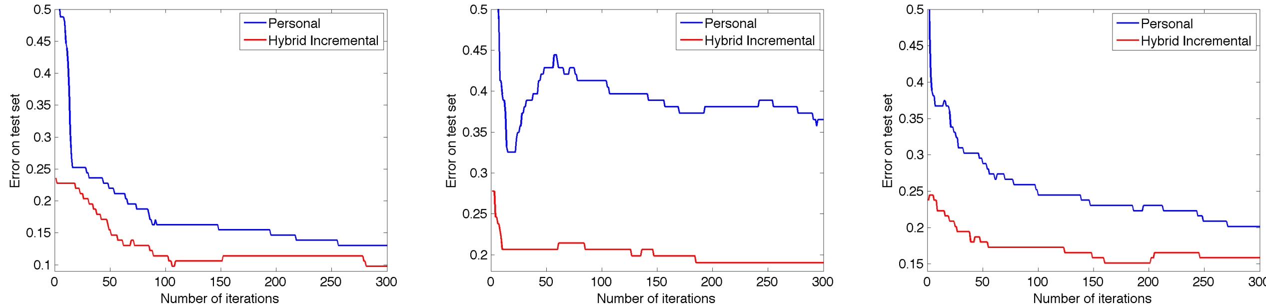


Figure 5: A comparison of the learning curves between the Personal method and the HI method. The x-axis shows the number of iterations of learning and the y-axis shows the test accuracy . From the Activity Prediction Data, three randomly selected users' results are presented. The HI method outperforms the Personal method, with better initialization and lower errors.

stairs class from the 6th user's incremental training data. As a result, upstairs class becomes the major predictions based on the cost-sensitive setting.

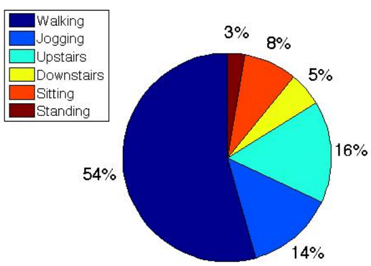


Figure 6: The imbalanced class distribution of training data of 12 test users from the Activity Prediction Data, which are randomly sampled.

6. CONCLUSION

In this paper, we propose a novel hybrid incremental (HI) method for activity recognition. Traditionally, we can either train models on a impersonal dataset or on a personal dataset. Our method can effectively combine the advantages of two approaches. After obtaining a model on impersonal dataset, the mobile devices can further apply incremental learning on the model using personal data. We focus on logistic regression model for its several benefits, including its small model size that saves bandwidth, good performance in activity recognition, and easy incremental update. We address two important problems that are likely to arise in practical implementations of this incremental learning task. The first problem is associated with extreme user diversity making it very difficult to tune learning-rate for each user. The second issue is related with personal data being so imbalanced at times that it may spoil the impersonal model. To overcome those problems, we applied an adaptive learning rate and cost sensitive technique. Finally the experimental results are used to validate our solutions.

Acknowledgment

This material is based, in part, upon work supported by the National Science Foundation under Grant No. 1344768.

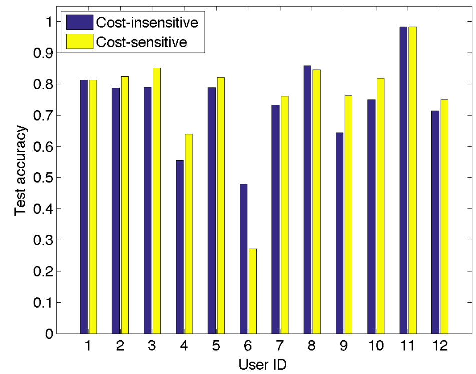


Figure 7: A comparison between the cost-insensitive model and the cost-sensitive model. The x-axis shows different test user's ID and the y-axis shows the test accuracy on each user's data. The cost-sensitive model performs better or very similar on 10 among 12 test users.

7. REFERENCES

[1] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, 150:304–317, 2015.

[2] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6):643–659, 2011.

[3] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *European conference on machine learning*, pages 39–50. Springer, 2004.

[4] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[5] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 437–442, 2013.

[6] A. Bifet and R. Gavalda. Learning from time-changing data with adaptive windowing. In *SDM*, volume 7,

pages 443–448. SIAM, 2007.

[7] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[8] W.-S. Chin, Y. Zhuang, Y.-C. Juan, and C.-J. Lin. A learning-rate schedule for stochastic gradient methods to matrix factorization. In *Advances in Knowledge Discovery and Data Mining*, pages 442–455. Springer, 2015.

[9] H. Du, A. Venkatakrishnan, G. M. Youngblood, A. Ram, and P. Pirolli. A group-based mobile application to increase adherence in exercise and nutrition programs: A factorial design feasibility study. *JMIR mHealth and uHealth*, 4(1), 2016.

[10] H. Du, G. M. Youngblood, and P. Pirolli. Efficacy of a smartphone system to support groups in behavior change programs. In *Proceedings of the Wireless Health 2014 on National Institutes of Health*, pages 1–8. ACM, 2014.

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[12] J. Gao, W. Fan, J. Han, and S. Y. Philip. A general framework for mining concept-drifting data streams with skewed distributions. In *SDM*, pages 3–14. SIAM, 2007.

[13] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM, 2011.

[14] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[16] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[17] 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 *Proceedings of the 13th international conference on Ubiquitous computing*, pages 355–364. ACM, 2011.

[18] P. Laskov, C. Gehl, S. Krüger, and K.-R. Müller. Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research*, 7:1909–1936, 2006.

[19] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *Pervasive Computing*, pages 1–16. Springer, 2006.

[20] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region newton method for logistic regression. *The Journal of Machine Learning Research*, 9:627–650, 2008.

[21] J. W. Lockhart and G. M. Weiss. The benefits of personalized smartphone-based activity recognition models. In *Proc. SIAM International Conference on Data Mining (SDM)*, pages 614–622, 2014.

[22] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1–7. IEEE, 2010.

[23] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

[24] I. A. H. Muller. Practical activity recognition using gsm data. In *Proceedings of the 5th International Semantic Web Conference (ISWC). Athens*, pages 1–8. Citeseer, 2006.

[25] K. Partridge and B. Begole. Activity-based advertising techniques and challenges. In *Proc. Workshop on Pervasive Advertising*, 2009.

[26] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.

[27] T. Schaul, S. Zhang, and Y. LeCun. No more pesky learning rates. *ICML (3)*, 28:343–351, 2013.

[28] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano. A comparative study of data sampling and cost sensitive learning. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 46–52. IEEE, 2008.

[29] K. Veropoulos, C. Campbell, N. Cristianini, et al. Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI*, pages 55–60, 1999.

[30] G. Wu and E. Y. Chang. KBA: kernel boundary alignment considering imbalanced data distribution. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):786–795, 2005.

[31] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang. Big data small footprint: The design of a low-power classifier for detecting transportation modes. *Proceedings of the VLDB Endowment*, 7:1429–1440, 2014.

[32] M. D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[33] M. Zeng, X. Wang, L. T. Nguyen, P. Wu, O. J. Mengshoel, and J. Zhang. Adaptive activity recognition with dynamic heterogeneous sensor fusion. In *2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE)*, pages 189–196. IEEE, 2014.

[34] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, pages 116–123. ACM, 2004.

[35] Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. Distributed newton methods for regularized logistic regression. In *Advances in Knowledge Discovery and Data Mining*, pages 690–703. Springer, 2015.