Large-Scale Personalized Human Activity Recognition Using Online Multitask Learning

Xu Sun, Hisashi Kashima, and Naonori Ueda, *Senior Member*, *IEEE*

Abstract—Personalized activity recognition usually has the problem of highly biased activity patterns among different tasks/persons. Traditional methods face problems on dealing with those conflicted activity patterns. We try to effectively model the activity patterns among different persons via casting this personalized activity recognition problem as a multitask learning issue. We propose a novel online multitask learning method for large-scale personalized activity recognition. In contrast with existing work of multitask learning that assumes fixed task relationships, our method can automatically discover task relationships from real-world data. Convergence analysis shows reasonable convergence properties of the proposed method. Experiments on two different activity data sets demonstrate that the proposed method significantly outperforms existing methods in activity recognition.

Index Terms—Multitask learning, online learning, human activity recognition, conditional random fields, data mining

1 Introduction

Acceleration sensor-based activity recognition is useful in practical applications [1], [2], [3], [4]. For example, in medical programmes, researchers hope to track lifestyle-related diseases. The traditional way of tracking, which requires manual operations, is ineffective both in time and accuracy. In sensor-based activity recognition, an accelerometer is employed (e.g., attached on the wrist of people) to automatically capture the acceleration signals in the daily life of counselees/patients, and the corresponding categories of activities can be automatically identified.

Although there was a considerable literature on activity recognition, most of the prior work discussed activity recognition in predefined limited environments [1], [2], [3]. It is unclear whether the previous methods can perform well in real-life environments. For example, most of the prior work assumed the beginning and ending time(s) of each activity are known beforehand, and the constructed recognition system only need to perform simple classifications of activities [1], [2], [3]. However, this is not the case for real-life activity sequences, where different activities are continuous in a sequence. For example, people may first walk, then take a taxi, then take an elevator, and the boundaries of the activities are unknown beforehand. An example of real-life activities with continuous sensor signals is shown in Fig. 1.

- *X. Sun is with the Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, Beijing, China, and the School of EECS, Peking University, No.5 Yiheyuan Road, Haidian District, Beijing, China. E-mail: xusun@pku.edu.cn.*
- *H. Kashima is with the Department of Mathematical Informatics, The University of Tokyo, Hongo, Tokyo, Japan. E-mail: kashima@mist.i.u-tokyo.ac.jp.*
- *N. Ueda is with the NTT Communication Science Laboratories, Kyoto, Japan. E-mail: ueda@cslab.kecl.ntt.co.jp.*

Manuscript received 19 Oct. 2011; revised 14 Aug. 2012; accepted 26 Oct. 2012; published online 13 Dec. 2012.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-10-0645.
Digital Object Identifier no. 10.1109/TKDE.2012.246.

More importantly, to the best of our knowledge, there is *no* previous work that systematically studied personalized activity recognition. Because of the difficulty of collecting training data for activity recognition, most of the prior work simply merge all personal data for training. We will show in our experiments that simply merging the personal data for training an activity recognizer will result in quite weak performance. Due to the fact that different persons usually have very different activity patterns, it is natural to construct one personalized model for each person. However, the new problem is the data sparseness of personalized activity recognition, because usually each person only has very limited amount of labeled training data.

To deal with the first problem of continuous activities, we propose structured classification methods for continuous activity recognition. To deal with the second problem of personalized learning, we apply the idea of multitask learning where each task corresponds to a specific person in activity recognition. We will propose an *online multitask learning* method for *personalized and continuous* activity recognition. The major motivation for using online training algorithm is to speed up the training. Since multitask learning is computationally expensive, it is important to speed up the training with online algorithms. We will show the proposed method is fast and scalable on large-scale data, which involves massive number of persons. We will also show reasonable convergence properties of the proposed method. Finally, we will perform experiments on two different data sets to demonstrate the superiority of the proposed method over existing methods.

This journal paper is substantially different compared with the previous work [5]. First, the methodology is different. The proposed method is more efficient than the method in [5], by using accelerated multitask learning based on task-similarity approximations and probabilistic sampling. Second, the convergence analysis on the proposed method is new. Third, the methodology and experiments on dealing with completely new tasks in the test stage is new. Fourth, the experiments on the Bao04 data

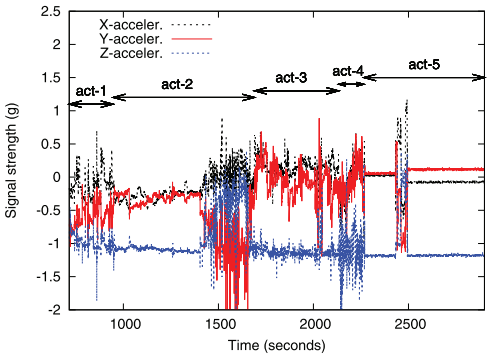


Fig. 1. An example of real-life continuous activities draw from our data, in which the corresponding 3D acceleration signals are collected from attached sensors. In this figure, "g" is the magnitude of *acceleration of gravity*.

set are new. There are also many other differences on the methods, analysis, experiments, and presentation.

2 Related Work and Motivations

First, we will introduce related work on activity recognition [1], [2], [3], [6], [7], [8]. Then, we will review multitask learning and highlight the novelties of this work. After that, we will briefly review conditional random fields (CRFs) and online training, which are related to our work.

2.1 Activity Recognition

Most of the prior work on activity recognition treated the task as a single-label classification problem [1], [2], [3]. Given a sequence of sensor signals, the activity recognition system predicts a single label (representing a type of activity) for the whole sequence. Ravi et al. [3] used decision trees (DTs), support vector machines (SVMs) and K-nearest neighbors (KNNs) models for classification. Bao and Intille [1] and Pärkkä et al. [2] used decision trees for classification. A few other works treated the task as a structured classification problem. Huynh et al. [4] tried to discover latent activity patterns by using a Bayesian latent topic model (Bayesian LTM). Hodges et al. [9] presented a work in activity monitoring for assisting the people with cognitive impairments. Most recently, Sun et al. [6] used CRFs and latent conditional random fields for activity recognition.

To our knowledge, there is only very limited work on the study of personalized activity recognition. A major reason is that most of the previous studies contain only a few participants. The limited number of participants is inadequate for a reliable study of personalized activity recognition. For example, in [3], the data were collected from two persons. In [4], the data were collected from only one person. In [2], the data were collected from 16 persons. Because of the difficulty of collecting training data, most of the prior work simply merge all personalized data for training. Personalized activity recognition and how to solve data sparseness in personalized activity recognition were not adequately studied. Table 1 summarizes prior work on activity recognition.

2.2 Multitask Learning

Multitask learning has been the focus of much interest in machine learning societies over the last decade. Traditional multitask learning methods include: sharing hidden nodes in neural networks [10], [11]; feature augmentation among interactive tasks [12]; producing a common prior in hierarchical Bayesian models [13], [14]; sharing parameters or common structures on the learning or predictor space [15], [16]; multitask feature selection [17]; a convex formulation that learns the similarities between tasks [18]; and matrix regularization-based methods [19], [20], among others.

Recent development of multitask learning is online multitask learning, started from [21]. Dekel et al. [21] assume that the tasks are related by a global loss function and the goal is to reduce the overall loss via online algorithm. With a similar but somewhat different motivation, Bartlett et al. [22] and [23] studied alternate formulations of online multitask learning under traditional expert advice models. This is a formulation to exploit low-dimensional common representations [24], [25]. Online multitask learning is also considered via reducing mistake bounds [26].

Our multitask learning proposal will be substantially different from the existing work of online multitask learning. While Yang et al. [17] focused on multitask feature selection and Agarwal et al. [23] focused on online matrix regularization, our proposal relates to neither feature selection nor matrix regularization. We will propose a tighter combination of online learning and multitask learning, with a new objective function and a novel training method. In addition, compared to the existing work of online multitask learning, a significant novelty of our method is that this method can adaptively learn task relationships (i.e., task similarities) from data, rather than using prior knowledge for defining a fixed task-relationship matrix, as presented in [26].

TABLE 1
Summarizing Prior Accelerometer-Based Activity Recognition Studies

	Models	Continuous activity?	Personalized learning?
Bao [1]	DTs	×	Limited
Pärkkä [2]	DTs	×	×
Ravi [3]	DTs, SVMs	×	×
Huynh [4]	Bayesian LTM	√	×
Sun [6]	CRFs, LCRFs	√	×
Sun [5]	Multi-Task Learner	√	√
This Work	Improved Multi-Task Learner	√	√

2.3 Conditional Random Fields

Conditional random fields are very popular models for structured classification [27]. Assuming a feature function that maps a pair of observation sequence $\boldsymbol{x}$ and label sequence $\boldsymbol{y}$ to a global feature vector $\boldsymbol{f}$, the probability of a label sequence $\boldsymbol{y}$ conditioned on the observation sequence $\boldsymbol{x}$ is modeled as follows [27]:

$$P(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \frac{\exp\left[\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x})\right]}{\sum_{\forall \boldsymbol{y}'} \exp\left[\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}', \boldsymbol{x})\right]}, \tag{1}$$

where $\boldsymbol{w}$ is a parameter vector.

Following the traditional setting on CRFs [28], the global feature vector is a summation of the local feature vectors on each observation node. The local feature vectors are defined based on the activity recognition problem, which include node features based on a single label and transition features based on label transitions. The node features and transition features are then mapped into a feature vector. After that, the global feature vector is summed over the local vectors of features.

Typically, computing $\sum_{\forall \boldsymbol{y}'} \exp[\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}', \boldsymbol{x})]$ could be computationally intractable. This summation can be computed using dynamic programming techniques [27]. To make the dynamic programming techniques applicable, the dependencies of labels are chosen to obey the Markov property. This has a computational complexity of $O(NK^M)$. N is the length of the sequence; K is the dimension of the label set; M is the length of the Markov order used by local features.

Given a training set consisting of n labeled sequences, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, for $i = 1 \ldots n$, parameter estimation is performed by maximizing the objective function

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n} \log P(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{w}) - R(\boldsymbol{w}). \tag{2}$$

The first term of this equation represents a conditional log-likelihood of a training data. The second term is a regularizer for reducing overfitting. We employed an L_2 prior, $R(\boldsymbol{w}) = \frac{\|\boldsymbol{w}\|^2}{2\sigma^2}$. In what follows, we denote the conditional log likelihood of each sample, $\log P(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w})$, as $\ell(i, \boldsymbol{w})$. The final objective function is as follows:

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n} \ell(i, \boldsymbol{w}) - \frac{\|\boldsymbol{w}\|^2}{2\sigma^2}. \tag{3}$$

2.4 Stochastic Gradient Descent (SGD)

There are two major approaches for training a log-linear model: batch training and online training. Batch training methods include, for example, steepest gradient descent, conjugate gradient descent (CG), and limited-memory BFGS (LBFGS) [29]. In such training methods, gradients are computed by using all training instances. Therefore, typically, the weight update process is quite slow.

To speed up the training, people turn to online training methods. A representative online training method is the stochastic gradient descent [30], [31]. The SGD uses a small randomly drawn subset of the training samples to approximate the gradient of the objective function given by (3). In this way, one can update the model parameters much more frequently and speed up the convergence. Suppose $\hat{\mathcal{S}}$ is a randomly drawn subset of the full training set $\mathcal{S}$, the stochastic objective function is then given by

$$\mathcal{L}_{stoch}(\boldsymbol{w}, \hat{\mathcal{S}}) = \sum_{i \in \mathcal{S}} \ell(i, \boldsymbol{w}) - \frac{|\hat{\mathcal{S}}|}{|\mathcal{S}|} \frac{\|\boldsymbol{w}\|^2}{2\sigma^2}.$$

The extreme case is a batch size of 1, and it gives the maximum frequency of updates, which we adopt in this work. In this case, $|\hat{\mathcal{S}}| = 1$ and $|\mathcal{S}| = n$ (suppose the full training set contains n samples). In this case, we have

$$\mathcal{L}_{stoch}(\boldsymbol{w}, \hat{\mathcal{S}}) = \ell(i, \boldsymbol{w}) - \frac{1}{n} \frac{\|\boldsymbol{w}\|^2}{2\sigma^2}, \tag{4}$$

where $\hat{\mathcal{S}} = \{i\}$. The model parameters are updated in such a way:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \gamma_k \nabla_{\boldsymbol{w}_k} \mathcal{L}_{stoch}(\boldsymbol{w}, \hat{\mathcal{S}}), \tag{5}$$

where k is the update counter, γ_k is the learning rate. A typical convergent choice of learning rate can be found in [32]:

$$\gamma_k = \frac{\gamma_0}{1 + k/N},$$

where γ_0 and N are two constants. This scheduling guarantees ultimate convergence [30], [31].

3 A New Multitask Learning Framework

In this section, we introduce the multitask learning framework. For every positive integer q, we define $\mathcal{N}_q = \{1, \ldots, q\}$. Let T be the number of tasks (number of persons in activity recognition) which we want to simultaneously learn. For each task $t \in \mathcal{N}_T$, there are n data examples $\{(\boldsymbol{x}_{t,i}, \boldsymbol{y}_{t,i}) : i \in \mathcal{N}_n\}$ available. The signals are encoded in a text file using a sequence of numbers to indicate the timing and strength of the three-axis signals.

In practice, the number of examples per task may vary but we have kept it constant for denotational simplicity. We use $\boldsymbol{D}$ to denote the $n \times T$ matrix whose tth column is given by the vector $\boldsymbol{d}_t$ of data examples. In other words, $\boldsymbol{d}_t$ consists of data samples of the t'th person. For example, if the multitask learning contains 10 persons, there will be 10 data sample vectors, $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{10}$. Putting together the 10 data sample vectors will lead to the $n \times 10$ matrix $\boldsymbol{D}$.

3.1 Model

Our goal is to learn the vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$ from the data $\boldsymbol{D}$. For denotational simplicity, we assume that each of the weight vectors is of the same size f (feature dimension), and corresponds to the same ordering of features. We use $\boldsymbol{W}$ to denote the $f \times T$ matrix whose tth column is given by the vector $\boldsymbol{w}_t$. For example, if the multitask learning contains 10 persons, then there will be 10 weight vectors, $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{10}$, corresponding to the 10 persons, respectively. Putting together the 10 weight vectors will lead to the $f \times 10$ matrix $\boldsymbol{W}$.

We learn $\boldsymbol{W}$ by maximizing the objective function

$$\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D}) \triangleq \mathrm{Likelihood}(\boldsymbol{W}, \boldsymbol{D}) - R(\boldsymbol{W}), \tag{6}$$

where $\text{Likelihood}(\boldsymbol{W}, \boldsymbol{D})$ is the cumulative likelihood on the tasks, namely,

$$\text{Likelihood}(\boldsymbol{W}, \boldsymbol{D}) = \sum_{t \in \mathcal{N}_T} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D}), \tag{7}$$

and $\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D})$ is defined as follows:

$$\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D}) \triangleq \sum_{t' \in \mathcal{N}_T} \left[\alpha_{t,t'} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right]. \tag{8}$$

$\alpha_{t,t'}$ is a real-valued *task similarity* between two tasks, with $\alpha_{t,t'} = \alpha_{t',t}$ (symmetric). Intuitively, a task similarity $\alpha_{t,t'}$ measures the *similarity* between the tth task and the t'th task. For example, in activity recognition, $\alpha_{t,t'}$ estimates the *similarity* of the activity patterns between the person t and the person t'. $\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})$ is defined as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'}) &\triangleq \sum_{i \in \mathcal{N}_n} \log P(\boldsymbol{y}_{t',i} \mid \boldsymbol{x}_{t',i}, \boldsymbol{w}_t) \\ &= \sum_{i \in \mathcal{N}_n} \ell_{t'}(i, \boldsymbol{w}_t), \end{aligned} \tag{9}$$

where $P(\cdot)$ is a prescribed probability function. In this paper, we use the CRF probability function, (1). The second step is just a simplified denotation by defining $\ell_{t'}(i, \boldsymbol{w}_t) \triangleq \log P(\boldsymbol{y}_{t',i} \mid \boldsymbol{x}_{t',i}, \boldsymbol{w}_t)$.

Finally, $R(\boldsymbol{W})$ is a regularization term for dealing with overfitting. In this paper, we simply use L_2 regularization:

$$R(\boldsymbol{W}) = \sum_{t \in \mathcal{N}_T} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}. \tag{10}$$

To summarize, our multitask learning objective function is as follows:

$$\text{Obj}(\boldsymbol{W}, \boldsymbol{D}) = \sum_{t,t' \in \mathcal{N}_T} \left[\alpha_{t,t'} \sum_{i \in \mathcal{N}_n} \ell_{t'}(i, \boldsymbol{w}_t)\right] - \sum_{t \in \mathcal{N}_T} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}.$$

To simplify denotation, we introduce a $T \times T$ matrix $\boldsymbol{A}$, such that $\boldsymbol{A}_{t,t'} \triangleq \alpha_{t,t'}$. We also introduce a $T \times T$ functional matrix $\boldsymbol{\Phi}$, such that $\boldsymbol{\Phi}_{t,t'} \triangleq \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})$. Then, the objective function can be compactly expressed as follows:

$$\text{Obj}(\boldsymbol{W}, \boldsymbol{D}) = \text{tr}(\boldsymbol{A}\boldsymbol{\Phi}^\top) - \sum_{t \in \mathcal{N}_T} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}, \tag{11}$$

where tr means *trace*. In the following content, we will first discuss a simple case that the task-similarity matrix $\boldsymbol{A}$ is fixed. After that, we will focus on the case that $\boldsymbol{A}$ is unknown.

3.2 Learning with Fixed Task Similarities

The original problem of multitask learning essentially consists of two subproblems: learning weight vectors (based on fixed task similarities) and learning task similarities. The problem of learning weight vectors based on fixed task similarities should be studied because this is a crucial subproblem of the original problem. With fixed task similarities, the optimization problem is as follows:

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\text{argmax}} \left[\text{tr}(\boldsymbol{A}^*\boldsymbol{\Phi}^\top) - \sum_{t \in \mathcal{N}_T} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}\right]. \tag{12}$$

It is clear to see that we can independently optimize $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$ when $t \neq t'$. In other words, we can independently optimize each column of $\boldsymbol{W}$, and therefore derive the optimal weight matrix $\boldsymbol{W}^*$. For $\boldsymbol{w}_t$ (i.e., the t'th column of $\boldsymbol{W}$), its optimal form is

$$\boldsymbol{w}_t^* = \underset{\boldsymbol{w}_t}{\text{argmax}}\, \psi(\boldsymbol{w}_t, \boldsymbol{D}), \tag{13}$$

where $\psi(\boldsymbol{w}_t, \boldsymbol{D})$ has the form as follows:

$$\psi(\boldsymbol{w}_t, \boldsymbol{D}) = \sum_{t' \in \mathcal{N}_T} \left[\alpha_{t,t'}^* \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right] - \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}. \tag{14}$$

This optimization problem is a cost-sensitive optimization problem. We present a cost-sensitive online training algorithm, called *online multitask learning with fixed task similarities (OMT-F)*, for this optimization. The OMT-F algorithm is shown in Fig. 2. In the OMT-F learning algorithm, the update term $\boldsymbol{g}_t$ is derived by weighted sampling over different tasks. The weighted sampling is based on fixed task similarities, $\boldsymbol{A}^*$. The update term $\boldsymbol{g}_t$ has a form as follows:

$$\boldsymbol{g}_t = \sum_{t' \in \mathcal{N}_T} \left[\boldsymbol{\alpha}_{t,t'}^* \nabla_{\boldsymbol{w}_t} \ell_{t'}(i_{t'}, \boldsymbol{w}_t)\right] - \frac{1}{n} \nabla_{\boldsymbol{w}_t} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2}, \tag{15}$$

where $\boldsymbol{\alpha}_{t,t'}^* = \boldsymbol{A}_{t,t'}^*$ and $i_{t'}$ indexes a random sample selected from $\boldsymbol{d}_{t'}$. Then, the expectation (over distribution of data) of the update term is as follows:

$$\begin{aligned} \text{E}(\boldsymbol{g}_t) &= \sum_{t' \in \mathcal{N}_T} \left\{ \boldsymbol{\alpha}_{t,t'}^* \left[\frac{1}{n} \nabla_{\boldsymbol{w}_t} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right] \right\} - \frac{1}{n} \nabla_{\boldsymbol{w}_t} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2} \\ &= \frac{1}{n} \left\{ \sum_{t' \in \mathcal{N}_T} \left[\boldsymbol{\alpha}_{t,t'}^* \nabla_{\boldsymbol{w}_t} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right] - \nabla_{\boldsymbol{w}_t} \frac{\|\boldsymbol{w}_t\|^2}{2\sigma_t^2} \right\} \\ &= \frac{1}{n} \nabla_{\boldsymbol{w}_t} \psi(\boldsymbol{w}_t, \boldsymbol{D}). \end{aligned}$$

3.2.1 Convergence Analysis

The possible convergence result for stochastic learning is the *"almost sure convergence"*: to prove that the stochastic algorithm converges toward the solution with probability 1 [30]. To prove the weight matrix $\boldsymbol{W}$ produced by algorithm OMT-F converges to the maximum $\boldsymbol{W}^*$ of (11), we need to guarantee the convexity of the loss function (i.e., $-\text{Obj}(\boldsymbol{W}, \boldsymbol{D})$). Since $-\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})$ is convex for every valid t and t', the most simple way to make sure the convexity of the loss function is to bound the task similarities. If the task-similarity matrix $\boldsymbol{A}$ is a nonnegative real-valued matrix (i.e., $\boldsymbol{A}_{t,t'} \geq 0$ for $\forall t, \forall t'$), the loss functions $-\text{Obj}(\boldsymbol{W}, \boldsymbol{D})$ and $-\psi(\boldsymbol{w}_t, \boldsymbol{D})$ are guaranteed to be convex.

If a stochastic update is convergent, it means that either the gradients or the learning rates vanish near the optimum [33]. According to [33], it is reasonable to assume that the variance of the stochastic gradient does not grow faster than the norm of the real gradient itself. Also, it is reasonable to assume that $\|\boldsymbol{g}_t\|^2$ behaves

Algorithm Learning with *fixed* task-similarities (OMT-F)
Input: $\boldsymbol{W} \leftarrow \boldsymbol{0}, \boldsymbol{D}, \boldsymbol{A}^*$
for $t \leftarrow 1$ to T
. **for** 1 to *convergence*
. . **for** 1 to n
. . . $\boldsymbol{g}_t \leftarrow -\frac{1}{n}\nabla_{\boldsymbol{w}_t}\frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}$
. . . **for** $t' \leftarrow 1$ to T
. . . . Draw $i \in \mathcal{N}_n$ at random
. . . . $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_t + \boldsymbol{A}^*_{t,t'}\nabla_{\boldsymbol{w}_t}\ell_{t'}(i, \boldsymbol{w}_t)$
. . . $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_t + \gamma \boldsymbol{g}_t$
Output: $\forall t$, $\boldsymbol{w}_t$ converges to $\boldsymbol{w}_t^*$; i.e., $\boldsymbol{W}$ converges to $\boldsymbol{W}^*$.

Algorithm Learning with *unknown* task-similarities (OMT-U)
Input: $\boldsymbol{W} \leftarrow \boldsymbol{0}, \boldsymbol{D}, \boldsymbol{A}$
for 1 to *empirical convergence*
. $\boldsymbol{W} \leftarrow$ OMT-F$(\boldsymbol{W}, \boldsymbol{D}, \boldsymbol{A})$
. **for** $t \leftarrow 1$ to T
. . **for** $t' \leftarrow 1$ to T
. . . Update $\boldsymbol{A}_{t,t'}$ using Eq. (19) or Eq. (18)
Output: $\boldsymbol{A}$ empirically converges to $\hat{\boldsymbol{A}}$. $\boldsymbol{W}$ converges to $\hat{\boldsymbol{W}}$.

Algorithm Accelerated OMT Learning with unknown task-similarities (OMT)
Input: $\boldsymbol{W} \leftarrow \boldsymbol{0}, \boldsymbol{D}, \boldsymbol{A}$, m, q
for 1 to m
. SGD training on $\boldsymbol{W}$ with single-task setting
for $t \leftarrow 1$ to T
. **for** $t' \leftarrow 1$ to T
. . Update $\boldsymbol{A}_{t,t'}$ using Eq. (19) or Eq. (18)
for m+1 to *convergence*
. **for** $t \leftarrow 1$ to T
. . **for** 1 to *convergence*
. . . **for** 1 to integer(n/q)
. . . . $\boldsymbol{g}_t \leftarrow -\frac{1}{n}\nabla_{\boldsymbol{w}_t}\frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}$
. . . . **for** $t' \leftarrow 1$ to T
. . . . . Draw $i \in \mathcal{N}_n$ at random
. . . . . $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_t + q\boldsymbol{A}^*_{t,t'}\nabla_{\boldsymbol{w}_t}\ell_{t'}(i, \boldsymbol{w}_t)$
. . . . $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_t + \gamma \boldsymbol{g}_t$
Output: $\boldsymbol{A}$ was approximated by $\hat{\boldsymbol{A}}$. $\boldsymbol{W}$ converges to $\hat{\boldsymbol{W}}$ based on $\hat{\boldsymbol{A}}$.

Fig. 2. Online multitask learning algorithms (using batch size of 1). The derivation of $\frac{1}{n}$ before the regularization term was explained in (4).

quadratically within the final convergence region. Both assumptions can be expressed as follows:

$$\boldsymbol{E}(\|\boldsymbol{g}_t\|^2) \leq a + b\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2, \tag{16}$$

where $a \geq 0$ and $b \geq 0$. Based on the assumptions, the *convergence theorem* has been given [33]: Two conditions on the learning rate are sufficient conditions for the *almost sure convergence* of the stochastic learning to optimum. The two conditions on the learning rate are as follows [33]:

$$\sum \gamma_k = \infty \ \ and \ \ \sum \gamma_k^2 < \infty. \tag{17}$$

With those preparations, there is the convergence result for the OMT-F:

Theorem 1. *Given the assumption that $\boldsymbol{A}$ is a nonnegative real-valued matrix, and the assumptions of (16) and (17), the parameters $\boldsymbol{W}$ produced by the OMT-F online learning algorithm are "almost surely convergent" toward the maximum $\boldsymbol{W}^*$ of (11).*

The proof can be derived by extending the proof of [33].

3.3 Learning with Unknown Task Similarities

For many practical applications, the task similarities are hidden variables that are unknown. To solve this problem, we present a heuristic learning algorithm to learn the task similarities and optimize model weights in an alternative optimization manner. In this case, the learning problem includes not only model weights, but also the hidden task similarities.

Our alternative learning algorithm with unknown task similarities, called OMT-U, is presented in the middle of Fig. 2. In the OMT-U learning, the OMT-F algorithm is

employed as a subroutine. In the beginning of the OMT-U, model weights $\boldsymbol{W}$ and task similarities $\boldsymbol{A}$ are initialized by a diagonal matrix with 1 on diagonal entries and 0 on other entries. $\boldsymbol{W}$ is then optimized to $\hat{\boldsymbol{W}}$ by using the OMT-F algorithm, based on the fixed $\boldsymbol{A}$. Then, in an alternative way, $\boldsymbol{A}$ is updated based on the optimized weights $\hat{\boldsymbol{W}}$. After that, $\boldsymbol{W}$ are optimized again based on updated (and fixed) task similarities. This iterative process continues until empirical convergence of $\boldsymbol{A}$ and $\boldsymbol{W}$.

In updating task similarities $\boldsymbol{A}$ based on $\boldsymbol{W}$, a natural idea is to estimate a task similarity $\alpha_{t,t'}$ based on the similarity between weight vectors, $\boldsymbol{w}_t$ and $\boldsymbol{w}_t$. The similarity between weight vectors can be calculated by using kernels, including the popular Gaussian RBF and polynomial kernels. We can define Gaussian RBF kernel to estimate similarity between two tasks:

$$\alpha_{t,t'} \triangleq \frac{1}{C}\exp\left(-\frac{\|\boldsymbol{w}_t - \boldsymbol{w}_{t'}\|^2}{2\sigma^2}\right), \tag{18}$$

where C is a real-valued constant for tuning the magnitude of task similarities. Intuitively, a big C will result in "weak multitasking" and a small C will make "strong multitasking." σ is used to control the variance of a Gaussian RBF function. Alternatively, we can use polynomial kernel (normalized and homogeneous) to estimate similarities between tasks:

$$\alpha_{t,t'} \triangleq \frac{1}{C}\frac{\langle \boldsymbol{w}_t, \boldsymbol{w}_{t'}\rangle^d}{\|\boldsymbol{w}_t\|^d \cdot \|\boldsymbol{w}_{t'}\|^d}, \tag{19}$$

where $\langle \boldsymbol{w}_t, \boldsymbol{w}_{t'}\rangle$ means inner product between the two vectors (i.e., $\boldsymbol{w}_t^\top \boldsymbol{w}_{t'}$); d is the degree of the polynomial kernel; $\|\boldsymbol{w}_t\|^d \cdot \|\boldsymbol{w}_{t'}\|^d$ is the normalizer; C is a real-value constant for controlling the magnitude of task similarities. Actually, the normalized polynomial kernel is natural and easy to understand. For example, when $d = 1$, the normalized kernel has exactly the form $\frac{1}{C}\cos\theta$, where θ is the angle between $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$ in the euclidean space.

3.4 Accelerated OMT Learning

3.4.1 Approximation on Task Similarities

The OMT learning algorithm can be further accelerated using approximate update of the task similarities, $\boldsymbol{A}$. The naive OMT-U learning algorithm waits for the convergence of the model weights $\boldsymbol{W}$ (in the OMT-F step) before updating the task similarities $\boldsymbol{A}$. In practice, we can update task similarities $\boldsymbol{A}$ before the convergence of the model weights $\boldsymbol{W}$, and no longer use the alternative style optimization. For example, we can update task similarities $\boldsymbol{A}$ after running only m passes of the OMT-F algorithm with diagonal matrix A (OMT-F with diagonal A equals to the SGD training with single-task setting). This can avoid the repeated update of the task similarities. This can bring a much faster training speed of the OMT method. In practice, our experiments demonstrate that even set $m = 1$ can produce good approximation on the true task similarities.

3.4.2 Probabilistic Sampling

In addition, we accelerate the training speed of the OMT method via probabilistic sampling over the related tasks. For example, suppose we use a probabilistic sampling factor of $1/q$. Suppose Tasks A and B have a task similarity of α. In training the weights of Task A, we can sample the training instances in Task B with the probability of $1/q$ and enlarge the gradient update factor to $q\alpha$, instead of using all training instances of Task B with the gradient update factor of α. In the original case, the gradient update factor of α is just from the task similarity. By using probabilistic sampling accelerations, the training speed will be accelerated because only a small portion of the training instances of Task B are used for computing gradients. It can be proved that the probabilistic sampling is a theoretically sound approximation, because the expectation of the gradient will not be changed in an asymptotic point of view and the asymptotic convergence analysis will still be sound.

Accelerated OMT learning is simply called OMT in this paper. The detailed accelerated OMT learning is presented in the bottom of Fig. 2. We will adopt this accelerated version of the OMT learning for experiments. In the experiment section, we will compare the (accelerated) OMT method with a variety of strong baseline methods.

3.4.3 Convergence Analysis

We will show in experiments that the polynomial kernel works better than the Gaussian RBF kernel. Hence, we will more focus on the polynomial kernel in this study. Since the polynomial kernel may make the loss function nonconvex, the convergence analysis is more difficult. We will show reasonable convergence properties by using polynomial kernels. Instead of showing that the model weights $\boldsymbol{W}$ converges, we show the loss function $-\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D})$ converges.

The convergence results rely on the following assumptions following [30], in addition to the assumptions of (16) and (17): First, the loss function $-\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D})$ is three times differentiable with continuous derivatives. The loss function has a lower bound that is nonnegative. Second, when the norm of the weights $\boldsymbol{w}_t$ is larger than a value D, $-\boldsymbol{w}_t \nabla_{\boldsymbol{w}_t} \psi(\boldsymbol{w}_t, \boldsymbol{D})$ is bounded:

$$\inf_{\boldsymbol{w}_t^2 > D} -\boldsymbol{w}_t \nabla_{\boldsymbol{w}_t} \psi(\boldsymbol{w}_t, \boldsymbol{D}) > 0. \tag{20}$$

Third, when the norm of the weights $\boldsymbol{w}_t$ is smaller than E with $E > D$, the norm of the update term is bounded:

$$\sup_{\boldsymbol{w}_t^2 < E} \|\boldsymbol{g}_t\| \leq K. \tag{21}$$

With those preparations, there is the following result for general online optimization with potentially nonconvex objective function:

Theorem 2. *Suppose the task similarities are updated by using the polynomial kernels, (19), and we have the assumptions of (16), (17), (20), and (21). For the OMT online learning algorithm, the loss function $-\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D})$ converges almost surely to $-\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D})^\infty$, and its gradient $-\nabla_{\boldsymbol{W}}\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D})$ converges almost surely to $\boldsymbol{0}$.*

The proof can be derived by extending the proof of [30] (stochastic learning on nonconvex objective functions). In other words, for the nonconvex loss functions, the algorithm will converge to *extremal points* of loss functions. Extremal points include local optimums.

TABLE 2
Features Used in the Activity Recognition Task

Single-axis based features:
(1) Signal strength features: $\{s_{i-2},\ s_{i-1},\ s_i,\ s_{i+1},\ s_{i+2},\ s_{i-1}s_i,\ s_is_{i+1}\} \times\{y_i,\ y_{i-1}y_i\}$
(2) Mean feature: $m_i \times\{y_i,\ y_{i-1}y_i\}$
(3) Standard deviation feature: $d_i \times\{y_i,\ y_{i-1}y_i\}$
(4) Energy feature: $e_i \times\{y_i,\ y_{i-1}y_i\}$
Multi-axis based features:
(1) Correlation features: $\{c_{1,2,i}, c_{2,3,i}, c_{1,3,i}\} \times\{y_i,\ y_{i-1}y_i\}$

$\mathcal{A}\times\mathcal{B}$ means a Cartesian product between two sets; i represents the window index; y_i and $y_{i-1}y_i$ represent CRF label and label-transition. Since the single-axis based features on the three axes are extracted in the same way, for simplicity, we only describe the features on one axis. For multiaxis-based features, we use 1, 2, and 3 to index/represent the three axes.

We have shown reasonable convergence properties of polynomial kernels. On the other hand, although the Gaussian RBF kernel works worse on empirical experiments, it has a theoretical advantage that $0 \le \alpha_{t,t'} \le \frac{1}{C}$, and therefore, $\boldsymbol{A}$ will always be a nonnegative matrix. In this case, convergence analysis is simpler, and Theorem 1 can be applied. The loss function is convex, and the OMT-F step guarantees to find global optimum of the loss function. A good thing is that our multitask learning framework can flexibly choosing polynomial kernel or Gaussian RBF kernel.

4 Experiments on ALKAN Data

We use the ALKAN data set [34] for experiments. This data set contains about 4×10^6 samples in temporal sequences. We extracted the personalized data sets of 20 individuals from the ALKAN data. The data were collected by *iPod/iPhone* accelerometers with the sampling frequency of 20 Hz. A sample contains four values: time stamp and triaxial signals. For example, {539.266(s), 0.091(g), -0.145(g), -1.051(g)}, in which "g" is the *acceleration of gravity*. There are activity labels including *"walking/running," "on elevator/escalator," "taking car/bus/train," "standing/sitting/discussing,"* and so on.

Following [6], we randomly selected 85 percent of samples for training, 5 percent samples for development data, and the rest 10 percent samples for testing. Following [6], the evaluation metric is sample-window accuracy (the number of correctly predicted sample-windows divided by the total number of sample-windows). Other evaluation metrics, like precision and recall, tend to be misleading in this task, because an activity segment is typically long, and small difference on the boundaries of segments can cause very different precision and recall. On the other hand, the accuracy metric is more reliable in this scenario.

4.1 Feature Engineering

Following prior work in activity recognition [1], [2], [3], [4], we use acceleration features, mean features, standard deviation, energy, and correlation features. The features are listed in Table 2. Following previous work [1], [2], window-based features (mean, energy, deviation, etc.) are extracted by using a window size of 128. We use exactly the same feature set for all systems. Taking Fig. 1, for example, the first window consists of the first signal sample until the 128th signal sample, and mean features and standard deviation will be extracted from this window of signals samples. Since the sampling frequency is 20 Hz, each signal sample will be 1/20 second. Then, the second window consists of the 129th signal sample until the 256th signal sample, and features will be extracted in this window. This process goes on in a similar way.

We denote the window index as i. The mean feature is simply the averaged signal strength in a window:

$$m_i = \frac{\sum_{k=1}^{|w|} s_k}{|w|},$$

where $s_1, s_2, \ldots$ are the signal magnitudes in a window. The energy feature is defined as follows:

$$e_i = \frac{\sum_{k=1}^{|w|} s_k^2}{|w|}.$$

The deviation feature is defined as follows:

$$d_i = \sqrt{\frac{\sum_{k=1}^{|w|}(s_k - m_i)^2}{|w|}},$$

where the m_i is the mean value defined before. The correlation feature is defined as follows:

$$c_{1,2,i} = \frac{covariance_{1,2,i}}{d_{1,i}d_{2,i}},$$

where $d_{1,i}$ and $d_{2,i}$ are the deviation values on the i'th window of the axis-1 and the axis-2, respectively. The $covariance_{1,2,i}$ is the covariance value between the i'th windows of the axis-1 and the axis-2. We defined correlation feature between other axis pairs in the same manner.

4.2 Experimental Setting

Baselines are adopted to make a comparison with the OMT method, including the SGD-Single training for each single person (using only this person's data for training), and the SGD-Merge training (merging all the training data of different persons to train a unified model).

In preliminary experiments, we find using $\sigma = 1$ and $d = 1$ worked well for Gaussian RBF and polynomial kernels. Therefore, we set $\sigma = 1$ for Gaussian RBF kernel and $d = 1$ for polynomial kernel. For the hyperparameter C, we test $C = 3{,}5{,}10{,}20$, and choose the optimal one. We will show detailed values of C in experimental results. The OMT, SGD-Single, and SGD-Merge methods use the same setting of decaying learning rate. Experiments were performed on a computer with Intel(R) Xeon(R) 2.40-GHz CPU.

4.3 Results and Discussion

To study multitask learning with different scales, we perform experiments on 5-person, 10-person, and 20-person data in an incremental way. The experimental results are listed in Table 3. Since the proposed method and SGD baselines are randomized algorithms, we also show the standard deviations (*std*) over three repeated experiments. Due to the fact that continuous activity recognition requires activity recognition and boundary disambiguation at the same time, it is expected to be much more difficult than prior work on noncontinuous recognition. As we can see, the OMT method significantly outperformed baseline

TABLE 3
Results on the ALKAN Data of 5 Persons, 10 Persons, and 20 Persons

#Person = 5	Overall Acc. (%)	#Passes	Training Time (sec)
SGD-Merge	58.29 (±1.47)	60	519
SGD-Single	68.15 (±0.86)	40	344
OMT-RBF, C=10 (proposal)	**69.27** (±0.61)	40	749
OMT-Poly, C=10 (proposal)	68.93 (±0.34)	40	533
#Person = 10	Overall Acc.	#Passes	Train Time (sec)
SGD-Merge	61.67 (±0.92)	60	598
SGD-Single	69.03 (±1.39)	30	300
OMT-RBF, C=10 (proposal)	69.32 (±0.36)	30	1206
OMT-Poly, C=10 (proposal)	**72.84** (±0.46)	30	1211
#Person = 20	Overall Acc.	#Passes	Train Time (sec)
SGD-Merge	60.75 (±0.15)	70	2069
SGD-Single	62.53 (±0.71)	30	853
OMT-RBF, C=20 (proposal)	65.42 (±0.81)	30	6636
OMT-Poly, C=20 (proposal)	**66.55** (±1.30)	30	6541

OMT is the proposed method. SGD-Single is the personalized SGD training; SGD-Merge is the nonpersonalized SGD training: All the personal training data are merged for training.

training methods. On 5-person data, the OMT method outperformed all of the other methods with at least 1.12 percent in terms of accuracy. Also, the advantages are 3.81 and 4.02 percent in 10-person and 20-person cases. In all of the data sets, the superiority of the OMT method is significant over other methods. Moreover, from the incremental experiments, we can see that the OMT method scaled well with increased number of tasks. With the increase of the number of tasks, the superiority of the OMT method is more significant over the baseline methods. We did not observe prior reports of multitask learning on equal or more than 20 tasks.

Note that the *overall* accuracies of 5-person, 10-person, and 20-person data sets are not directly comparable to each other, simply because the data sets are different. For example, the 20-person data set contains the newly added 15 persons (compared with the 5-person data set), and the newly added 15 persons may have more noisy data. Nevertheless, the personal accuracies for specific persons are comparable among different scales.

4.3.1 Overall and Personal Curves

In Fig. 3, we show the accuracy curves by varying the number of training passes. Not only overall curves, but also personal ones are shown. From the overall curves, we can clearly see the superiority of the OMT method over other methods in different scales. We can see the personal curves are very diversified, and simply merging their data for unified SGD training is frustrating: The SGD-Merge method gave low accuracies and slow convergence speed. As can be seen, the OMT method is an ideal solution for this diversified situation.

The baseline methods had poor performance mainly because they cannot properly take use of the data from other tasks. The SGD-Single had poor performance because its own training data for each task are very limited, and the training data from other tasks cannot be employed. The

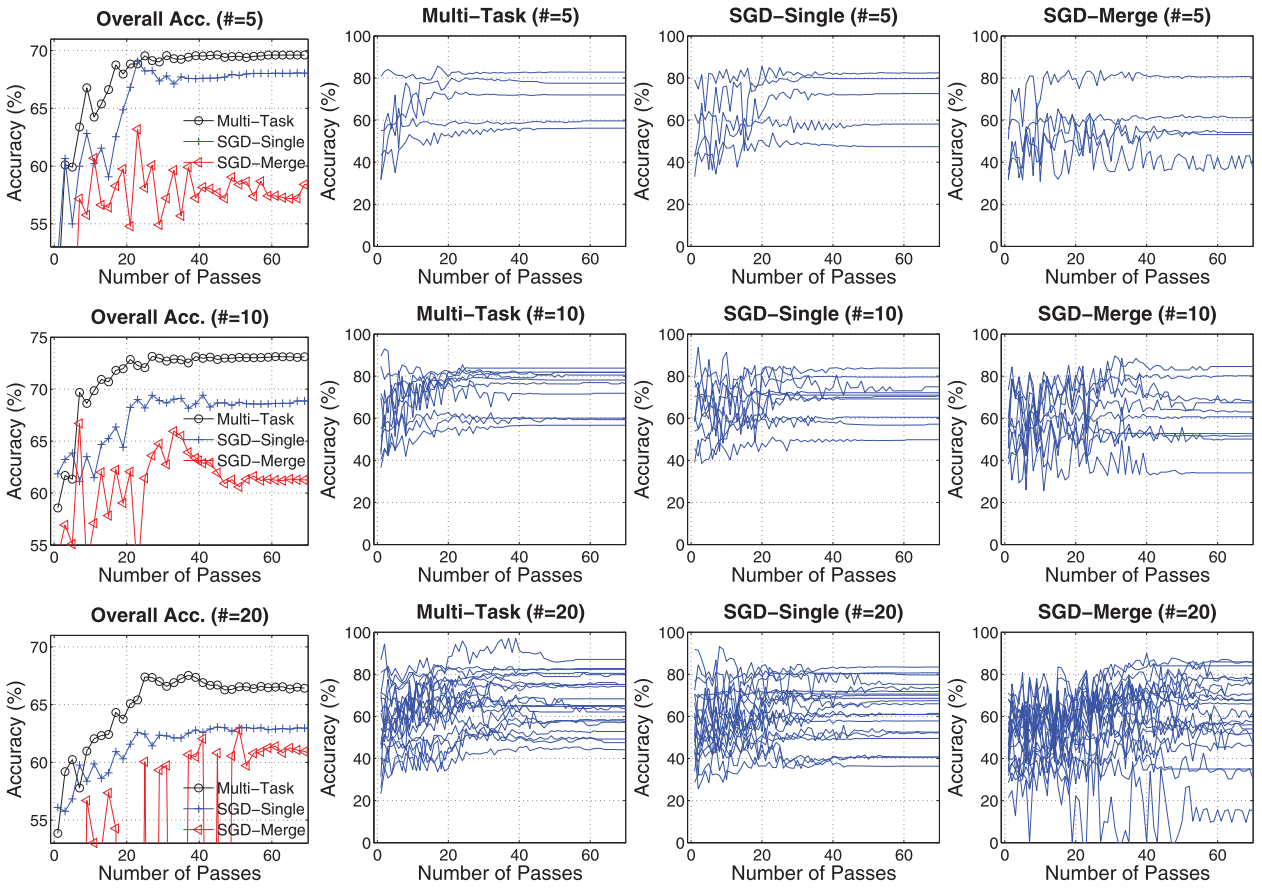


Fig. 3. Overall accuracies (averaged accuracies over all persons) and personal accuracies. (Top) Curves of 5 persons. (Middle) Curves on the 10-person data. (Bottom) Curves on the 20-person data. (First column) Overall curves. (Second column) Personal curves of the OMT method. (Third column) Personal curves of the SGD-Single. (Fourth column) Personal curves of the SGD-Merge.

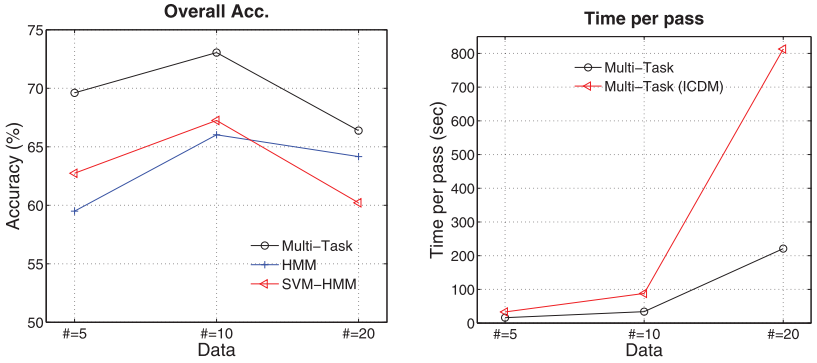


Fig. 4. (Left) Comparing the proposed method with HMM and SVM-HMM methods. (Right) Comparing the proposed method with the multitask learning method in [5].

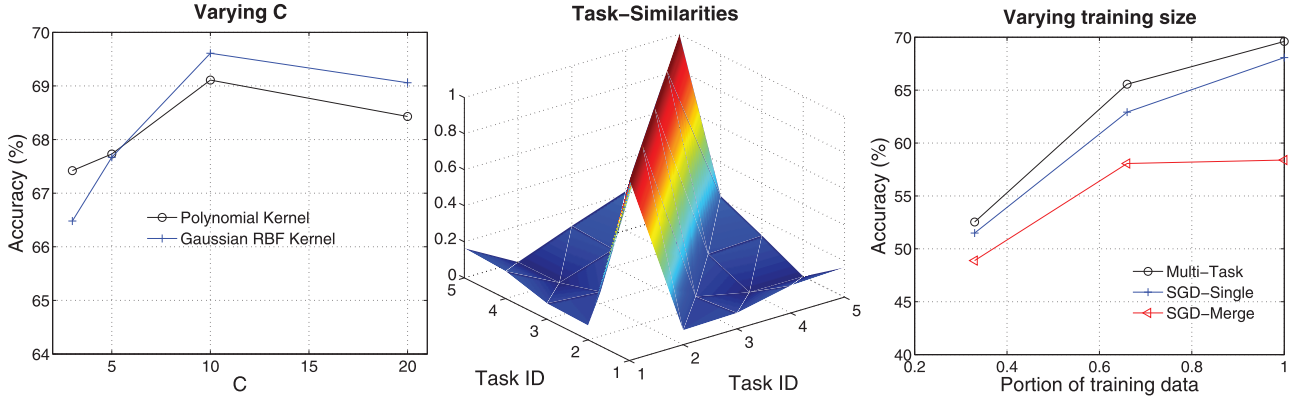


Fig. 5. (Left) Overall accuracy curves by tuning C. (Middle) OMT task similarities. The value on (i, j) corresponds to $\alpha_{i,j}$. (Right) Accuracies by varying the size of the training data. For simplicity, the figures are based on $\# = 5$.

inadequate training data led to poorly trained model parameters. For another baseline, SGD-Merge, it employed the data from other tasks, but it still had poor or even worse performance. The main reason was that the SGD-Merge cannot address the confliction from the biased data of other tasks. This can be observed from the severe fluctuations on accuracy curves of SGD-Merge. The severe fluctuations on accuracy curves reflected the conflictions among the training data.

The OMT method had improved performance because it can take use of the training data from other tasks and at the same time address the conflictions from the biased data of other tasks. As can be seen, the OMT method has higher accuracy than the SGD-Single because the OMT can take use of the additional data from other tasks. In addition, the OMT method has much smoother accuracy curves than the SGD-Merge because it can address the conflictions from the biased data of other tasks.

Hidden Markov models (HMM) are popular in many existing applications. In Fig. 4 (left), we also show the results based on hidden Markov models. In the plot, *HMM* represents the discriminatively trained hidden Markov model [35]. The HMM model is trained by the averaged perceptron algorithm presented in [35]. The *SVM-HMM* represents the SVM^{HMM} model[1] [36]. The SVM-HMM trains discriminative hidden Markov models using the SVM formulation. For both the HMM and the SVM-HMM models, there are five hidden states. As we can see from the figure, the proposed method consistently outperformed the HMM and SVM-HMM methods in all of the data sets.

In Fig. 4 (right), we compare the proposed method with the existing work in [5] in terms of training speed. The training speed is evaluated by *time per pass*, i.e., the wall-clock time used in each training iteration. As we can see, the proposed method is substantially more efficient compared with the existing work in [5]. In the $\# = 20$ data, the proposed method is about four times faster than the method in [5].

4.3.2 *Varying C, Task Similarities, Training Size*

In Fig. 5 (left), we make comparisons between polynomial and Gaussian RBF kernels and show overall accuracy curves by tuning the hyperparameter C of the kernels. As we can see, the RBF kernel worked better than the polynomial kernel on this data set. For both of the RBF and polynomial kernels, the setting of $C = 10$ worked the best.

Since the task similarities are unknown in this task, it will be interesting to check the learned task similarities based on the OMT method. In Fig. 5 (middle), we show the distribution of the task similarities on the 5-person data. The distribution is over the task pairs. Hence, the distribution is in a 3D space and the distribution of task similarities is symmetric. In addition, we observed that all task similarities were nonnegative.

In Fig. 5 (right), we show experimental results by varying the size of the training data. As we can see, the proposed method consistently outperformed baselines on different sizes of the training data.

1. http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html.

4.4 Discussion: Completely New Tasks

As a multitask learning method, following previous studies on multitask learning, our current setting is based on the assumption that the tasks in the test stage are not completely new. This means that a task has some data for training and some data for testing. This is a common setting in most of the previous studies on multitask learning. Here, we also consider a more difficult scenario that a task in the test stage is unobserved at all in the training stage. This scenario is more complicated and prior work on this topic is limited.

Suppose we have i tasks in the multitask training stage, $(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{w}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2, \boldsymbol{w}_2), \ldots, (\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{w}_i)$, where $\boldsymbol{x}_i$ is the vector of observation sequences of the i'th task, $\boldsymbol{y}_i$ is the vector of gold-standard label sequences of the i'th task, and $\boldsymbol{w}_i$ is the trained weight vector. Suppose $\boldsymbol{x}$ is the vector of observation sequences of a completely new task in the test stage that was unobserved in the training stage. Now, we need to calculate the $\boldsymbol{y}$ in the test stage.

Our solution is simple. The system calculates the task-similarity between the new task and the existing tasks in the training stage, and the most similar task in the training stage is picked to classify the new task in the test stage. First, the system calculates the similarities, $s_1, s_2, \ldots, s_i$, between the observation vector $\boldsymbol{x}$ and the observation vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i$ in the training stage. The derivation of s_i is based on the cosine similarity between two vectors:

$$s_i = \frac{\langle \bar{f}(\boldsymbol{x}_i), \bar{f}(\boldsymbol{x}) \rangle}{\|\bar{f}(\boldsymbol{x}_i)\| \cdot \|\bar{f}(\boldsymbol{x})\|},$$

where $\bar{f}(\boldsymbol{x})$ means the averaged feature vector of the observations $\boldsymbol{x}$. Then, the system computes the voted model weight vector $\boldsymbol{w} = \sum_1^i (s_i \boldsymbol{w}_i)$, and uses $\boldsymbol{w}$ to classify the new task. We call this method as *OMT with similarity-based decoding (OMT-SBD)*.

We performed experiments to evaluate this simple method. We randomly selected the data of 10 persons among the 20 persons for training, and the remaining 10 persons for testing. The data of the 10 persons in the testing stage are unobserved in the training stage. Our OMT-SBD method achieved the averaged accuracy of 49.23 percent (accuracy per task: 35.52, 37.24, 69.32, 62.84, 40.03, 56.58, 69.82, 55.60, 41.94, 23.35 percent). The baseline was the traditional method with merged training, which trained one unified weight vector for all testing data. The baseline method achieved the averaged accuracy of 46.96 percent (accuracy per task: 34.87, 35.17, 72.85, 55.29, 46.18, 44.91, 82.95, 44.59, 35.27, 17.54 percent). As we can see, the OMT-SBD method was more accurate than the baseline method in dealing with new tasks unobserved in training.

5 Experiments on Bao04 Data

To justify the proposed method, we perform experiments on another well-known data set: the Bao04 activity recognition data set [1].[2] The Bao04 data are very different from the ALKAN data. While the ALKAN data used one triaxial sensor, the Bao04 data used five biaxial sensors (attached on four limb positions plus right hip) to collect accelerations, with 10 acceleration signals overall. The data contain 2.6×10^6 samples in total. The acceleration signals were sampled at 76.25 Hz, which is more frequent than the 20 Hz sampling in ALKAN data.

The Bao04 data were collected from 20 persons. In [1], the activity recognition is studied in a noncontinuous manner. To conduct continuous activity recognition, we reformat the original data into continuous activity sequences via segmenting and connecting activities.

TABLE 4
Results on the Bao04 Data of 5 Persons, 10 Persons, and 20 Persons

#Person = 5	Overall Acc. (%)	#Passes	Training Time (sec)
SGD-Merge	87.86 (±0.32)	70	145
SGD-Single	90.81 (±0.27)	40	83
OMT-RBF, C=10 (proposal)	**91.16** (±0.09)	40	158
OMT-Poly, C=10 (proposal)	90.95 (±0.18)	40	158
#Person = 10	Overall Acc. (%)	#Passes	Training Time (sec)
SGD-Merge	89.28 (±0.24)	60	89
SGD-Single	90.98 (±0.08)	30	45
OMT-RBF, C=10 (proposal)	91.94 (±0.19)	30	153
OMT-Poly, C=10 (proposal)	**92.09** (±0.13)	30	141
#Person = 20	Overall Acc. (%)	#Passes	Training Time (sec)
SGD-Merge	88.79 (±0.07)	80	241
SGD-Single	91.26 (±0.21)	40	122
OMT-RBF, C=20 (proposal)	92.13 (±0.13)	40	754
OMT-Poly, C=20 (proposal)	**92.33** (±0.06)	40	737

5.1 Features and Experimental Settings

We use similar features from the previous experiment. The single-axis-based features are exactly the same to the previous experiment, as listed in Table 2. The multiaxis features are defined on the five axis pairs from the five biaxial sensors. Hence, there are five correlation features. Each correlation feature is defined in the same way as the ALKAN experiment. This feature set is also similar to the prior work [1] on this data set. Like the previous experiment, four baselines are adopted to make comparisons. Experimental settings are similar to the previous task on ALKAN data set. We avoid the redundant descriptions here.

5.2 Results and Discussion

The experimental results are listed in Table 4. Since the Bao04 data used five biaxial sensors to collect accelerations, the acceleration information is richer than the ALKAN data (which only used one triaxial sensor). As a result, the accuracies in this data are much higher than those in the ALKAN data. As we can see, the OMT method also achieved better accuracy than those baseline methods. Significance tests showed the OMT method is significantly better than other methods on both of the 10-person and 20-person data sets.

In Fig. 6, we show the curves of accuracies on varying the number of training passes. Compared with the ALKAN data, personal curves are less diversified in the Bao04 data. Nevertheless, similar to the ALKAN data, the OMT method performed more accurate than the SGD-Merge and SGD-Single.

In Fig. 7 (left), we show the learned task similarities among personal data pairs of the Bao04 data. As we can see, the task similarities were symmetric and highly diversified.

2. Data available at: http://architecture.mit.edu/house_n/data/Accelerometer/BaoIntilleData04.htm.

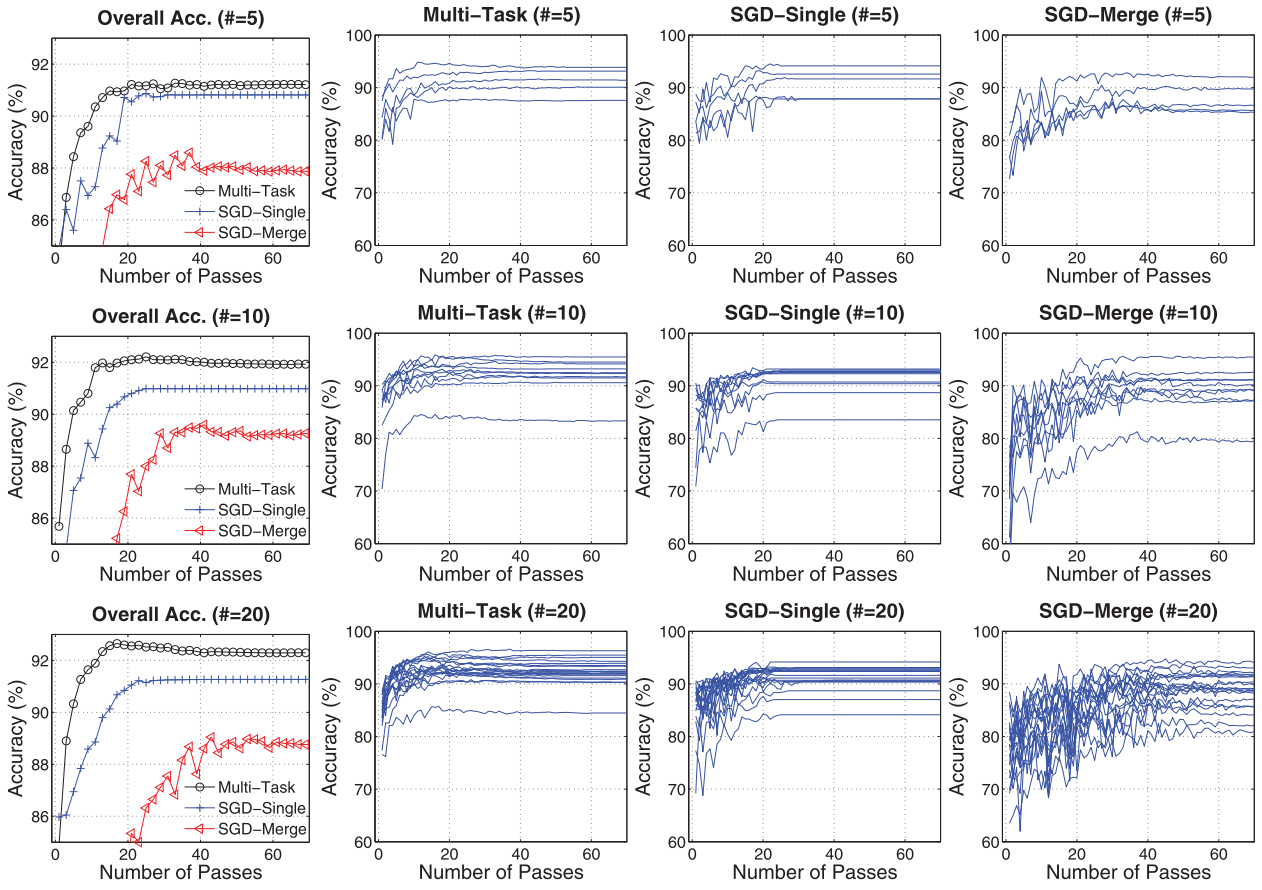


Fig. 6. Overall and personal accuracy curves on Bao04 data (20 persons). (First) Overall accuracies of different methods. (Second) Personal curves of the OMT method. (Third) Personal curves of SGD-Single. (Fourth) Personal curves of SGD-Merge.

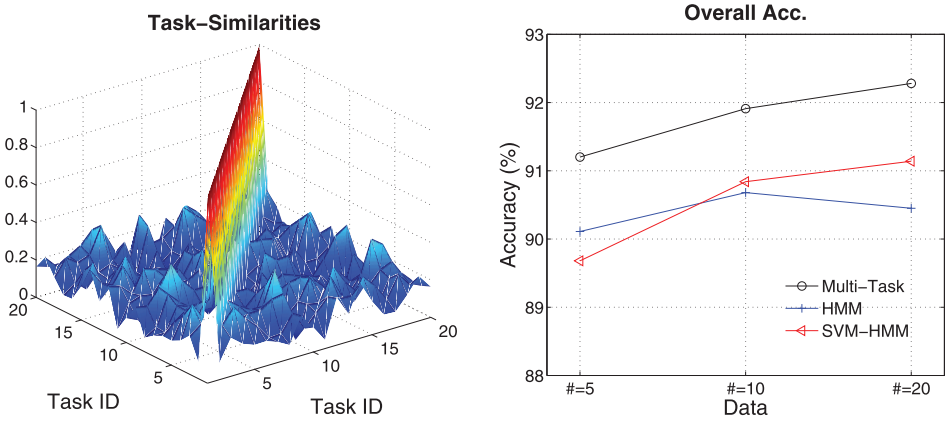


Fig. 7. (Left) OMT task similarities on Bao04 data. (Right) Comparing the proposed method with HMM and SVM-HMM methods.

Different task pairs have different task-similarity values. Similar to the ALKAN data, we observed that all task similarities were nonnegative.

In Fig. 7 (right), we show the results based on hidden Markov models, including discriminative HMM and SVM-HMM. As we can see, the proposed method consistently outperformed the HMM and SVM-HMM methods in all of the Bao04 data sets.

6 Conclusions and Future Work

In this paper, we studied personalized continuous activity recognition by using a new online multitask learning method. The proposed training method is fast and scalable for multitask learning with massive data in action recognition. Experiments on two different data sets demonstrated that the proposed method considerably outperformed existing baselines in activity recognition, and with fast training speed as well as sound scalability. We also learned interesting task similarities between tasks. The proposed method is a general technique, and it can be easily applied to other tasks. As future work, we plan to apply this method to other data mining tasks. We also plan to test some speech recognition methods (e.g., [37]) to activity recognition.

ACKNOWLEDGMENTS

This work was supported by the FIRST Program of JSPS, and National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101). The authors thank the reviewers who gave helpful comments. X. Sun is the corresponding author.

REFERENCES

[1] L. Bao and S.S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Proc. Int'l Conf. Pervasive Computing,* pp. 1-17, 2004.

[2] J. Präkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity Classification Using Realistic Data from Wearable Sensors," *IEEE Trans. Information Technology in Biomedicine,* vol. 10, no. 1, pp. 119-128, Jan. 2006.

[3] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, "Activity Recognition from Accelerometer Data," *Proc. Conf. Innovative Applications of Artificial Intelligence (IAAI '05),* pp. 1541-1546, 2005.

[4] T. Huynh, M. Fritz, and B. Schiele, "Discovery of Activity Patterns Using Topic Models," *Proc. 10th Int'l Conf. Ubiquitous Computing,* pp. 10-19, 2008.

[5] X. Sun, H. Kashima, R. Tomioka, N. Ueda, and P. Li, "A New Multi-Task Learning Method for Personalized Activity Recognition," *Proc. IEEE Int'l Conf. Data Mining (ICDM '11),* 2011.

[6] X. Sun, H. Kashima, T. Matsuzaki, and N. Ueda, "Averaged Stochastic Gradient Descent with Feedback: An Accurate, Robust, and Fast Training Method," *Proc. IEEE 10th Int'l Conf. Data Mining (ICDM '10),* pp. 1067-1072, 2010.

[7] J. Yin, D. Shen, Q. Yang, and Z.-N. Li, "Activity Recognition through Goal-Based Segmentation," *Proc. Nat'l Conf. Artificial Intelligence (AAAI '05),* M.M. Veloso and S. Kambhampati, eds., pp. 28-34, 2005.

[8] J. Yin, Q. Yang, and J. Pan, "Sensor-Based Abnormal Human-Activity Detection," *IEEE Trans. Knowledge Data Eng.,* vol. 20, no. 8, pp. 1082-1090, Aug. 2008.

[9] M.R. Hodges, M.W. Newman, and M.E. Pollack, "Object-Use Activity Monitoring: Feasibility for People with Cognitive Impairments," *Proc. AAAI Spring Symp.: Human Behavior Modeling,* pp. 13-18, 2009.

[10] J. Baxter, "A Model of Inductive Bias Learning," *CoRR,* vol. abs/1106.0245, informal publication, 2011.

[11] R. Caruana, "Multitask Learning," *Machine Learning,* vol. 28, no. 1, pp. 41-75, 1997.

[12] H. Daumé III, "Frustratingly Easy Domain Adaptation," *Proc. 45th Ann. Meeting of the Assoc. of Computational Linguistics,* pp. 256-263, June 2007.

[13] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian Processes from Multiple Tasks," *Proc. Int'l Conf. Machine Learning (ICML '05),* L.D. Raedt and S. Wrobel, eds., vol. 119, pp. 1012-1019, 2005.

[14] J. Zhang, Z. Ghahramani, and Y. Yang, "Learning Multiple Related Tasks Using Latent Independent Component Analysis," *Proc. Neural Information Processing System (NIPS '05),* 2005.

[15] N.D. Lawrence and J.C. Platt, "Learning to Learn with the Informative Vector Machine," *Proc. Int'l Conf. Machine Learning (ICML '04),* C.E. Brodley, ed., vol. 69, 2004.

[16] R.K. Ando and T. Zhang, "A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data," *J. Machine Learning Research,* vol. 6, pp. 1817-1853, 2005.

[17] H. Yang, I. King, and M.R. Lyu, "Online Learning for Multi-Task Feature Selection," *Proc. ACM Int'l Conf. Information and Knowledge Management,* pp. 1693-1696, 2010.

[18] Y. Zhang and D.-Y. Yeung, "A Convex Formulation for Learning Task Relationships in Multi-Task Learning," *Proc. Conf. Uncertainty in Artificial Intelligence (UAI '10),* pp. 733-742, 2010.

[19] A. Argyriou, C.A. Micchelli, M. Pontil, and Y. Ying, "A Spectral Regularization Framework for Multi-Task Structure Learning," *Proc. Neural Information Processing System (NIPS '07),* 2007.

[20] Y. Xue, D. Dunson, and L. Carin, "The Matrix Stick-Breaking Process for Flexible Multi-Task Learning," *Proc. 24th Int'l Conf. Machine Learning (ICML '07),* pp. 1063-1070, 2007.

[21] O. Dekel, P.M. Long, and Y. Singer, "Online Multitask Learning," *Proc. 19th Ann. Conf. Learning Theory (COLT),* G. Lugosi and H.-U. Simon, eds., pp. 453-467, 2006.

[22] J. Abernethy, P. Bartlett, and A. Rakhlin, "Multitask Learning with Expert Advice," *Proc. Ann. Conf. Learning Theory (COLT),* N.H. Bshouty and C. Gentile, eds., pp. 484-498, 2007.

[23] A. Agarwal, A. Rakhlin, and P. Bartlett, "Matrix Regularization Techniques for Online Multitask Learning," Technical Report UCB/EECS-2008-138, EECS Dept., Univ. of California, Berkeley, Oct. 2008.

[24] T. Evgeniou, C.A. Micchelli, and M. Pontil, "Learning Multiple Tasks with Kernel Methods," *J. Machine Learning Research,* vol. 6, pp. 615-637, 2005.

[25] P. Rai and H.D. Daumé III, "Infinite Predictor Subspace Models for Multitask Learning," *J. Machine Learning Research,* vol. 9, pp. 613-620, 2010.

[26] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear Algorithms for Online Multitask Classification," *Proc. Ann. Conf. Learning Theory (COLT),* R.A. Servedio and T. Zhang, eds., pp. 251-262, 2008.

[27] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. 18th Int'l Conf. Machine Learning (ICML '01),* pp. 282-289, 2001.

[28] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," *NAACL '03: Proc. Conf. North Am. Chapter of the Assoc. for Computational Linguistics on Human Language Technology,* pp. 134-141, 2003.

[29] J. Nocedal and S.J. Wright, *Numerical Optimization.* Springer, 1999.

[30] L. Bottou, "Online Algorithms and Stochastic Approximations," *Online Learning and Neural Networks,* D. Saad, ed., Cambridge Univ. Press, 1998.

[31] J.C. Spall, *Introduction to Stochastic Search and Optimization.* Wiley-IEEE, 2005.

[32] M. Collins, A. Globerson, T. Koo, X. Carreras, and P.L. Bartlett, "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks," *J. Machine Learning Research,* vol. 9, pp. 1775-1822, 2008.

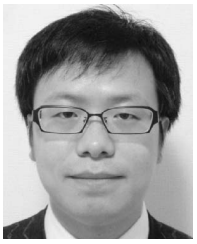
[33] L. Bottou, "Stochastic Learning," *Advanced Lectures on Machine Learning,* pp. 146-168, Springer, 2004.

[34] Y. Hattori, M. Takemori, S. Inoue, G. Hirakawa, and O. Sudo, "Operation and Baseline Assessment of Large Scale Activity Gathering System by Mobile Device," *Proc. DICOMO Workshop,* 2010.

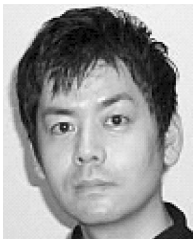
[35] M. Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP '02),* pp. 1-8, 2002.

[36] T. Joachims, T. Finley, and C.-N. Yu, "Cutting-Plane Training of Structural SVMs," *Machine Learning,* vol. 77, no. 1, pp. 27-59, 2009.

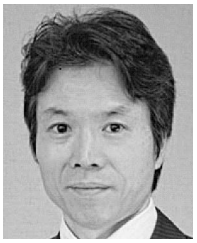
[37] M.J.F. Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language,* vol. 12, no. 2, pp. 75-98, 1998.



Xu Sun received the BE degree from the Huazhong University of Science and Technology in 2004, the MS degree from Peking University in 2007, and the PhD degree from the University of Tokyo in 2010. He is a "Bai Ren" professor in the School of EECS, Peking University. His research interests include natural language processing, data mining, and machine learning. He has worked on projects including machine translation, text information extraction, abbreviation processing, search query optimization, signal processing, discriminative latent variable models, online training, and multitask learning.



Hisashi Kashima received the BEng, MEng, and PhD degrees from Kyoto University in 1997, 1999, and 2007, respectively. He is an associate professor in the Department of Mathematical Informatics, University of Tokyo. Before joining the faculty, he was a research staff member in the Data Analytics Group of the Tokyo Research Laboratory at IBM Research during 1999-2009. His research interest includes machine learning and data mining.



Naonori Ueda received the BS, MS, and PhD degrees in communication engineering from Osaka University, Japan, in 1982, 1984, and 1992, respectively. In 1984, he joined the Electrical Communication Laboratories, NTT, Japan, where he was engaged in research on image processing, pattern recognition, and computer vision. In 1991, he joined the NTT Communication Science Laboratories, where he has invented a significant learning principle for optimal vector quantizer design, and invented statistical machine learning methods such as DAEM, SMEM algorithms, ensemble learning, semisupervised learning, variational Bayesian learning, and probabilistic multilabeled text model, and PMM. His current research interests include parametric and nonparametric Bayesian approach to machine learning, pattern recognition, data mining, signal processing, and cyber-physical systems. He has published more than 100 papers in his research areas and received many research awards. From 1993 to 1994, he was a visiting scholar at Purdue University, West Lafayette. He was the director of NTT Communication Science Laboratories, and is currently the director of the Machine Learning & Data Science Center of NTT. He is the subproject leader, Funding Program for World-Leading Innovative R&D on Science and Technology (First Program), Cabinet Office, government of Japan, March 2010-February 2014. He is a guest professor at the National Institute of Informatics. He has served as the program committee of SIGKDD since 2009. He is a fellow of the Institute of Electronics, Information, and Communication Engineers (IEICE) and a senior member of the IEEE.