Mobile Activity Recognition through Training Labels with Inaccurate Activity Segments

Takamichi TODA
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata, Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
5toda6@gmail.com

Sozo INOUE
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata, Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
sozo@mns.kyutech.ac.jp

Naonori UEDA
NTT Communication Science Laboratories
2-4 Hikaridai, Seika, Sagara-gun
Kyoto, 619-0237, JAPAN
ueda.naonori@lab.ntt.co.jp

ABSTRACT

In this paper, we propose an approach to improve mobile activity recognition, given a training dataset with inaccurate segments, in which the beginning and ending timestamps of homogeneous and continuous activities have inaccurate boundaries due to human errors. In the proposed approach, we A) convert the training dataset to multilabel samples, B) train the dataset by using a multilabel expectation maximization learning algorithm, and C) apply a segmentation method using not only the estimated labels but also the original segment information. We evaluate the proposed approach for three datasets, including simulation data and real activity data, two machine-learning algorithms, and various inaccuracies, and show that the proposed approach outperforms the naive methods as follows: 1) it fixes the segments of the training data and 2) improves the recognition accuracy through cross validation.

CCS Concepts

•**Human-centered computing** → *Mobile computing;*

Keywords

Activity recognition; mobile sensing; inaccurate segments; EM algorithm

1. INTRODUCTION

Recently, human activity recognition (HAR) through mobile sensor devices has been researched intensively, and its applications are anticipated in various fields such as sports and healthcare [2]. In activity recognition methods, supervised-machine learning is widely utilized with a collection of input sensor data associated with activity labels, which are used as the expected output data in the training (learning) phase. However, as the labels are often human-generated, it is possible that a segment, which may be the beginning and ending timestamps of homogeneous and continuous activities, has inaccurate boundaries caused by human errors. These inaccurate results decrease recognition accuracy.

In this paper, we propose an approach to improve the recognition accuracy, when inaccurate segments in the training dataset are given, by fixing the segments to maximize the recognition accuracy. In the proposed approach, A) we first convert each pair of input feature vectors and label to multilabeled samples. By this, we can consider the possible time lags of segments. B) Next, we train the dataset using a multilabel learning algorithm, which allows plural labels for each sample but assumes only one true label. The algorithm maximizes the likelihood of true labels by repeating the expectation maximization (EM) steps when the input feature vectors and multiple labels are given. Finally, as the estimated labels could be fragmented, we C) apply a segmentation method by using not only the estimated labels but also the original segment information. Thereby, the inaccurate segments in the training dataset are fixed; by using them for the training phase, the recognition accuracy of the new test data is improved through cross validation.

Next, we evaluated the proposed approach for three datasets, including simulation and real activity data, two machine-learning algorithms, and various inaccuracies. As a result, the proposed approach outperforms the naive methods in terms of 1) fixing the segments of the training data and 2) improving the recognition accuracy through cross validation, in any size of time lag, datasets, and machine-learning algorithms.

The contribution of our study is three-fold: 1) we proposed an approach to fix inaccurate activity segments, and evaluated it by using various datasets and machine-learning algorithms, 2) the approach was proved to improve recognition accuracy through evaluations by using various datasets and algorithms, and 3) it was demonstrated that the approach can also determine inaccurate segments in the existing open dataset.

2. PROBLEM DEFINITION

During the utilization of supervised machine learning for pattern recognitions, including mobile activity recognitions, the existence of inaccurate labels can prove problematic [12, 3]. For instance, in natural language processing, uncertain or missing labels reduce the recognition accuracy [5].

In addition, in the domain of activity recognition, we experience the problem of *inaccurate segments*, that is, the beginning and ending timestamps of homogeneous and continuous activities are different from the true values, as shown in Fig 1. These phenomena lead to inaccuracy of the trained model by supervised machine learning, as will be shown in Section 4.

MOBIQUITOUS '16, November 28-December 01, 2016, Hiroshima, Japan

ISBN 978-1-4503-4750-1/16/11...$15.00

DOI: http://dx.doi.org/10.1145/2994374.2994378

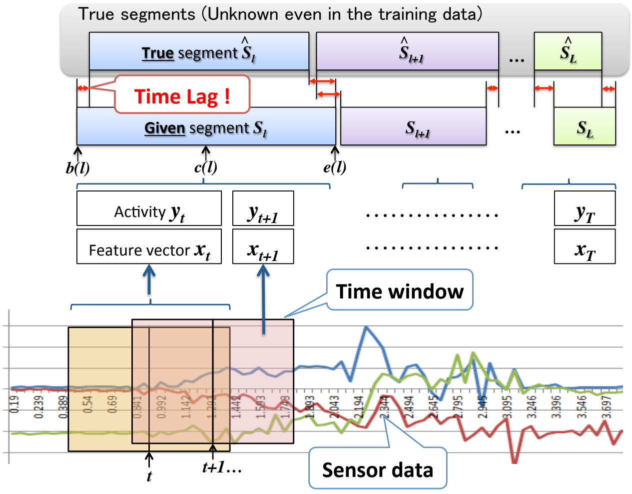


Figure 1: Problem illustration

When the labels are generated by humans, an online or offline method is considered. In online methods, the labels are generated in real time by an observer or the subject her/himself. The observer can record the timestamps on time; however, unintentional misses or latency is expected to some extent because of human factors. Moreover, if we cannot pay the costs for observers or if the activities are extremely private, the subject has to generate labels by self. However, it inevitably results in some latency because the subject requires some time to record timestamps.

In offline methods, accurate labels can be generated through accurate segmentations by recording the activities through videos or audios, and reviewing them later, although reviewing requires higher costs. Although, some activity changepoints, such as "Standing up from a chair and starting to walk," are ambiguous by nature, a method for improving activity recognition is required. Moreover, considering ubicomp applications, such as activities of daily living (ADL), the subjects often record their activities at the end of the days, and applying ideal offline methods, such as reviewing videos or audios, is not practical in many applications.

Therefore, considering that we cannot avoid inaccurate labels in ubicomp applications, we hypothesized our problem as follows:

Given the training dataset with inaccurate activity segments, the segments must be fixed to improve recognition accuracy.

2.1 Preliminary

As a preliminary step, we introduce the mathematical expressions used throughout this paper. Table 1 summarizes the expressions.

For simplicity, we assume that the time is expressed as an integer between 1 and T. We abbreviate the sequence $(1, 2, \cdots, T)$ as $1 : T$. For each t, we assume that a *feature vector*, consisting of several statistic values from the time window of the sensor input around t, is extracted. We specify the sequence of feature vectors $(x_1, x_2, \cdots, x_T)$ as $x_{1:T}$.

Moreover, A refers to the set of activity classes to be recognized. At each time t, we assume that the activity class label $y_t \in A$ is given. We define a *sample* at t as a feature value with its label (x_t, y_t).

Table 1: Basic expressions used in the paper

Symbol	Summary
A	The set of activity classes to be recognized
$1 : T$	The time sequence $(1, 2, \cdots, T)$.
x_t	The feature vector at time $t \in 1 : T$.
$x_{1:T}$	The sequence of feature vectors $(x_1, x_2, \cdots, x_T)$.
$y_t \in A$	The given activity label at time t.
$s_l := (a, b(l), e(l))$	The l'th given segment with activity $a \in A$ from $b(l) \in 1 : T$ to $e(l) \in 1 : T$ $(l \in 1 : L)$.
$\hat{s}_l := (a, \hat{b}(l), \hat{e}(l))$	The true segment for s_l .

We use the term *segment* as a continuous time range, where a homogenous activity is performed and represented as a pair of start and end times. By assuming that L segments appear in the sequence $y_{1:T}$, the l'th segment from time $b(l)$ to $e(l)$ with activity class a is defined as

$$s_l := (a, b(l), e(l)), \quad \text{where} \quad a \in A, 1 \leq b(l) \leq e(l) \leq T. \tag{1}$$

Although we can naturally derive $s_{1:L}$ when labels $y_{1:T}$ are given as training labels, the given segments have possibly different timestamps due to the problem of inaccurate segments introduced earlier. Therefore, we introduce the notation $\hat{s}_l := (a, \hat{b}(l), \hat{e}(l))$, which denotes the true segment for s_l.

2.2 Problem definition

Conventional activity recognition such as that proposed by Bao et al. [2] can be modeled as the problem for constructing function $y = f(x)$ from the given datasets $x_{1:T}$ and $y_{1:T}$ and estimating the activity label y'_t by

$$\arg_{y'_t} \max \mathbf{P}(y'_t | x'_t, f) \tag{2}$$

for a new input x'_t. Apparently, this does not consider activity sequences or segments.

In contrast, our goal can be represented as the problem to obtain the fixed segments $\hat{s}_{1:L}$ by

$$\arg_{\hat{s}_{1:L}} \max \mathbf{P}(\hat{s}_{1:L} | s_{1:L}, x_{1:T}). \tag{3}$$

for the training datasets $x_{1:T}$ and $s_{1:L}$, to construct function f from $x_{1:T}$ and the fixed $\hat{s}_{1:L}$, and to estimate the sequence $s'_{1:L'}$ by

$$\arg_{s'_{1:L'}} \max \mathbf{P}(s'_{1:L'} | x'_{1:T'}, f) \tag{4}$$

for a new feature sequence $x'_{1:T'}$.

3. PROPOSED APPROACH

In this section, we introduce the proposed approach for solving the problem of inaccurate segments in the training data for activity recognition. Figure 2 shows the processing flow of the proposed method. Moreover, Table 2 summarizes the additional mathematical expressions introduced in this section.

In our approach, we first

A) generate multilabels from the given labels according to the Gaussian distribution. assuming that the true labels exist in the center of the segment and becomes more unlikely for distant times.

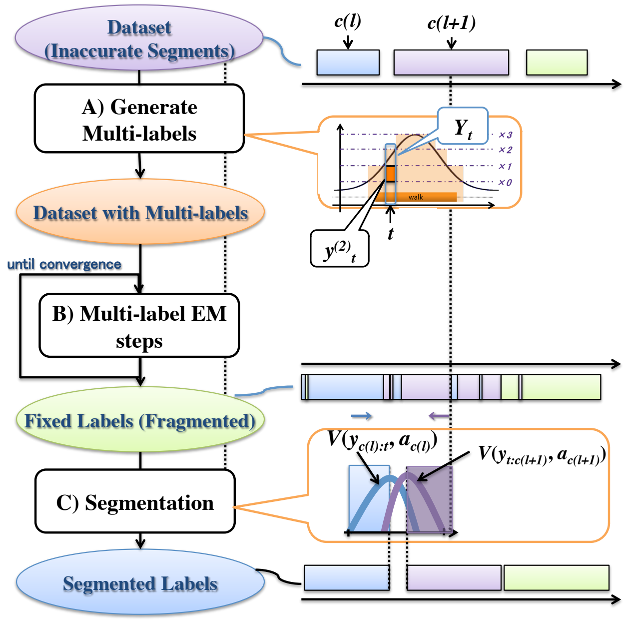


Figure 2: The overview of the proposed approach

Table 2: Additional expressions used in the proposed method

Symbol	Summary
$c(l) := \lfloor \frac{b(l)+e(l)}{2} \rfloor$	The center of the segment s_l.
$\tilde{y}_t \in A$	The estimated activity at time t.
$Y_t \subseteq A$	Set of labels at time t.
$m_t := \|\|Y_t\|\|$	Number of labels at time t.
D_t	Multilabel samples derived from x_t and Y_t
σ_l	Parameter for generating multiple labels
L_a	Set of segment numbers with activity class a.
Z_t	Normalization factor for time t.
$V(y_{t_1:t_2}, a)$	Number of samples with activity class $a \in A$ in $y_{t_1:t_2}$

This increases the possibility of true pairs x_t and y_t existing in the training dataset. Next, we

B) apply the probabilistic multilabel EM algorithm,

which estimates the most probable pairs of x_t and y_t from multiple possibilities.

Moreover, the learning method does not consider the data series, and we combine it with *segmentation* to estimate the boundaries of each segment. Therefore, we add a posterior process of

C) segmentation for the estimated labels,

which is original in the manner that it utilizes not only the activity labels passed from the previous step but also the original segment information.

3.1 A) Generating multi-labels

As input, we assume that the feature vectors $x_{1:T}$ and the segments $s_{1:L}$ are given.

First, we convert the given labels to be probabilistic so that each segment is distributed according to Gaussian distribution, in which the mean value is the center of the segment $c(l)$, and the standard deviation is proportional to the duration

$$\sigma_l := \alpha \cdot (e(l) - b(l)), \tag{5}$$

where α denotes a parameter which is a single constant throughout the approach. Therefore, according to the definition of Gaussian distribution $N(\mu, \sigma^2)$, the probability that segment s_l continues at time t is:

$$G(t, l) := N(c(l), \sigma_l^2) \tag{6}$$

Next, summing up for the same activity, we calculate

$$\mathbf{P}(y_t = a) := \frac{1}{Z_t} \sum_{l \in L_a} G(t, l) \tag{7}$$

for each time $t \in 1:T$ and each activity class $a \in A$, where L_a is the set of segment numbers with activity class a, defined as

$$L_a := \{l \in 1:L | y_{c(l)} = a)\}$$

and Z_t is a normalization factor for time t, defined as

$$Z_t := \sum_{l \in 1:L} G(t, l).$$

Moreover, by following Formula (7), we generate multiple labels for each time $t \in 1:T$. In other words, for each t, we duplicate the sample (x_t, a) to be proportional to $\mathbf{P}(y_t = a)$. We assume that the number of multilabels is represented by the constant m, that is, $m_t = m$ for any $t \in 1:T$ for simplicity. Thereby, we can duplicate sample (x_t, a) for

$$m \times P(a_t = a) \text{ times.}$$

Thus, the dataset has multiple labels for each time t, as described in the multilabel learning in the next step.

3.2 B) Probabilistic multi-label EM steps

Multilabel learning is a type of machine learning that allows plural labels for each learning sample [4, 17]. We focus on a special case of multilabel learning in which a sample may have plural labels but only one label is true; this case was introduced by Jin et al. [9]. Jin et al. formulated the algorithm by using the Kullback–Leibler divergence but we describe it through a practical procedure.

From the previous step, the following multilabels are obtained for each sample of time $t \in 1:T$

$$D_t = (x_t, y_t^{(1)}), (x_t, y_t^{(2)}), \cdots, (x_t, y_t^{(m_t)}), \tag{8}$$

By using these, we perform the following procedure:

1. (M-step) Construct the probability distribution $\mathbf{P}(y_t|x_t)$ through machine learning by using dataset $D_{1:T}$,

2. (E-step) By using model $\mathbf{P}(y_t|x_t)$, estimate the probabilities of activities $\mathbf{P}(\tilde{y}_t|x_t)$ using data $x_{1:T}$, and

3. generate multilabels again, that is, repeat (x_t, a) for $m \times \mathbf{P}(\tilde{y}_t|x_t)$ times for each $a \in A$. These data become the training data for the next step.

4. Repeat steps 1–3 until the results converge or their change becomes negligible, and

5. by using the conventional method in Formula (4), obtain the most likely activity $\tilde{y}_t$ for each x_t and $t \in 1:T$.

Because these steps are formulated as an EM algorithm, they converge to a local solution according to Jensen's inequality.

3.3 C) Segmentation

The predicted labels in the previous step could be fragmented. For recovering the segments, we need a method for segmentation. Here, we define *the amount of activity for duration* $t_1 : t_2$ as the number of samples with activity class a, defined as

$$V(\tilde{y}_{t_1:t_2}, a) := \sum_{t \in t_1:t_2} \delta(\tilde{y}_t = a)$$

where $\delta(y_t = a) := 1$ when $y_t = a$, and 0 otherwise.

By using this activity amount, we consider the likelihood of the duration being a part of the segment formulated as

$$\frac{V(\tilde{y}_{t_1:t_2}, a)}{t_2 - t_1}.$$

However, as the duration for segmentation between continuous activities is favorable to be longer, we multiply this by the length of the duration $t_2 - t_1$. Therefore, we only consider $V(\tilde{y}_{t_1:t_2}, a)$ to be proportional to the likelihood.

For each segment s_l in $l \in 1 : L$,

1. For t in $c(l) : c(l+1)$ between the centers of neighboring segments, calculate

$$V(\tilde{y}_{c(l):t}, a_{c(l)})$$

2. let the maximum likelihood point as the estimated boundary of the segment be defined as

$$\tilde{e}(l) \leftarrow \arg_{t \in c(l):c(l+1)} \max \{V(\tilde{y}_{c(l):t}, a_{c(l)})\}.$$

Here, for simplicity, we assume dummy segment boundaries as $e(0) = 1$ and $b(L+1) = T$.

We described the estimation of $e(l)$, which is the right boundary of the segment; however, this can be applied similarly for the left boundary $b(l)$ except that the calculation is applied toward the left between s_l and s_{l-1}.

Next, for each segment s_l in $l \in 1 : L$,

3. calculate $\tilde{b}(l)$.

After these calculations, some segmentations might be overlapped or have no labeled durations between the adjacent segments. In such a case, we consider their center as the boundary.

For a segment s_l, the following steps are used:

4. $\tilde{e}(l) \leftarrow \tilde{b}(l+1) \leftarrow \frac{\tilde{e}(l)+\tilde{b}(l+1)}{2}$.

We propose two versions of applying step 4:

EM + Sparse) must be applied only for the boundaries that overlap with the adjacent boundaries, that is, for $l \in 1 : L$, where $\tilde{b}(l+1) < \tilde{e}(l)$, and

EM + Dense) must be applied for all the boundaries of $l \in 1 : L$.

The sparse version obviously generates durations with no labels, regardless of whether the given labels have it. We evaluate and compare both versions in the next section.

4. EVALUATION

This section shows our evaluation of the proposed approach for several data sets, including simulation data and real activity datasets.

The research questions are as follows:

Question 1: Can the approach fix inaccurate segments?

Question 2: Can the fixed segments improve the recognition accuracy?

We examined these questions for three datasets, two machine-learning algorithms, and various time lags.

4.1 Dataset and Features

To evaluate our approach, we need an activity dataset consisting of a series of multiple activities. Therefore, we used three datasets: 1) a *simulation dataset* generated through a simulation and 2) *HAR dataset* and 3) *Human Activity Sensing Consortium (HASC) dataset* both of which have numbers of activity sequences collected by [1]. The *HASC* dataset also consists of data collected through several labs [10]. Table 3 lists the dataset details.

Table 3: Overview of the three datasets

Datasets→	Simulation	HAR	HASC
# of activity classes	6	6	6
# of sequences	30	30	9
# of segments in a sequence	$\mu = 4$ ($\sigma = 0$)	$\mu = 11.33$ ($\sigma = 1.84$)	$\mu = 14$ ($\sigma = 1.94$)
Segment durations [sec]	$\mu = 20$ ($\sigma = 0$)	$\mu = 32.88$ ($\sigma = 11.44$)	$\mu = 20.41$ ($\sigma = 18.20$)
Time window width [sec]	1	2.56	2
Time window shift [sec]	1	1.28	1
Dimensions of features	3	14	12

4.1.1 Simulation dataset

To control the number of segments and their durations, we generated a dataset through a simulation. The dataset consists of 6 activity classes and 30 activity sequences, each of which has four 20 s activity segments.

We assume that time windows are generated by splitting the sequence by 1 s, and each feature vector with three dimensions is calculated from a single time window.

Considering the values of the feature vectors, we assumed that the activity class generates discriminated values composed of 0 or 1, such as $[0, 0, 1]$ for the first class and $[0, 1, 0]$ for the second class, and added Gaussian random noises, with $\mu = 0$ and $\sigma = 0.1$.

4.1.2 UCI HAR dataset

For the open dataset with natural settings, we adopted the (*HAR*) dataset[1] [1]. The sensor data were collected using smartphones equipped with 3-axial accelerometers and gyroscopes. The smartphones were attached on the waists of 30 subjects. The subjects attended six types of activity classes: “Standing”, “Sitting”, “Laying”, “Walking”, “Walking downstairs”, and “Walking upstairs”.

Each of the sequences, recorded once for each subject, has 11.3 segments on an average ($\sigma = 1.84$), and the mean segment duration is 32.88 seconds ($\sigma = 11.44$).

Instead of the raw sensor data, the HAR dataset provides the feature values calculated from the time windows with 2.56 s width and 50

It originally provides 561 feature variables, each of which is derived from the statistic values; such as mean, energy, entropy, and coefficients; from the X, Y, and Z axes of either the time or frequency domains. We reduced the feature variables to 14 by applying stepwise-feature selection [7].

4.1.3 HASC dataset

From among the activity dataset collected during the HASC[2], we adopted nine activity sequences collected from different subjects. They attended six types of activity classes: “Standing”, “Walking”,

[1]https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

[2]http://hasc.jp/

"Skipping", "Running", "Walking downstairs", and "Walking upstairs". Each of the sequence has an average of 14 segments (with $\sigma = 1.94$), and the mean segment duration is 20.41 s ($\sigma = 18.20$).

We utilized time-window widths of 2 s, shifting them by 0.5 s, and calculated the following features for X, Y, and Z axes from each time window: mean values, variances, interquartile ranges, and the number of crosses of the mean value zones.

4.2 An Evaluation Method

To prepare various inaccuracies for the given segments $s_{1:L}$ in the dataset, we "intentionally" added time lags for the segments. For this, we randomly shifted the start and end times ($b(l)$ and $e(l)$, respectively) by following the Gaussian random noise $\mu = 0$ and varying σ. In addition, we avoided the boundaries to go beyond the neighboring boundaries, for example, to avoid $e(l) > b(l+1)$.

When applying the approach, we set the parameters $\alpha = 0.1$ and $m = 20$. Moreover, we stopped the repetition of the EM steps when the variation dropped under 5

For the evaluating the solution of Question 1, we used the mean absolute error of the segments by seconds. In addition, for solving Question 2, we used the accuracy measure of the rate of correctly-recognized samples among the total samples. These two measures are popular in activity or pattern recognition research. For simplicity, we omitted the samples originally labeled or classified as "others" in advance.

We used the statistical analysis software R[3] to analyze the results, and utilized the naive Bayes classifier or the RandomForest classifier to construct $\mathbf{P}(y_t|x_t)$ in the EM steps.

We describe the evaluation method for each question as follows.

4.2.1 Question 1: *Can the approach fix inaccurate segments?*

By assuming that we have a training dataset with inaccurate timestamps for the segments, we evaluated whether the timestamps are corrected. Because the goal is to fix the segments, we did not require applying cross validations, and were able to use only the training datasets with labels.

We compared the following methods:

Default) The time lags of the given segments. These are the same as the intentionally-generated noises through Gaussian distribution.

EM+Sparse) apply EM algorithms and sparse segmentation, and

EM+Dense) apply EM algorithms and dense segmentation.

Note that we omitted the EM algorithm without segmentation here because its output will be fragmented and will not be a one-to-one correspondence with the original labels.

4.2.2 Question 2: *Can the fixed segments improve the recognition accuracy?*

Similar to the standard evaluation of activity recognition, we first split the data into the training and test data, in terms of one-person (sequence)-leave-out cross validation. Next, for the training data, we fixed the segments by applying the proposed algorithm. Finally, we evaluated the recognition accuracy by applying it to the test data.

We compared the following methods:

Default) the traditional method as Formula (4),

EM) apply EM algorithms without any segmentation,

[3]http://www.r-project.org/

EM+Sparse) apply EM algorithms and sparse segmentation, and

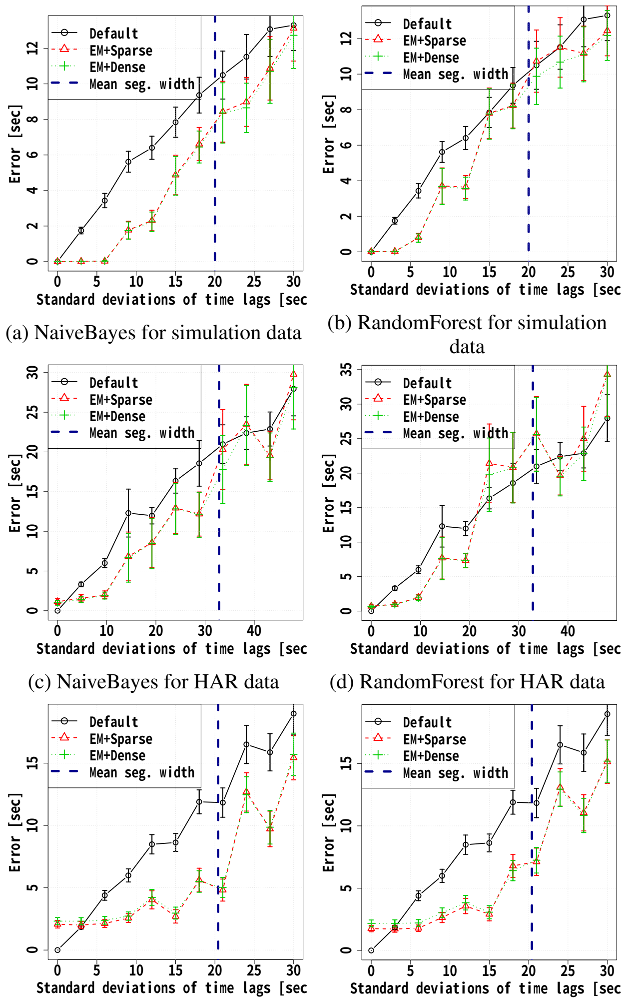
EM+Dense) apply EM algorithms and dense segmentation.

Note that our proposed approach is applied to the training phase, and therefore the test step is the same for these four methods.

4.3 Results

4.3.1 Question 1: *Can the approach fix inaccurate segments?*

Figure 3 shows the experimental results for Question 1. The horizontal axis denotes the standard deviations [seconds] in creating time lags for inaccurate segments, and the vertical axis denotes the mean absolute errors from the true segments. [%]. Each graph shows the types of evaluation data and the machine-learning algorithms for constructing $\mathbf{P}(y_t|x_t)$. In each graph, each line shows either **Default**, **EM+Sparse**, or **EM+Dense** method with the bar of $\pm 1\sigma$ among different activity sequences.



(a) NaiveBayes for simulation data (b) RandomForest for simulation data

(c) NaiveBayes for HAR data (d) RandomForest for HAR data

(e) NaiveBayes for HASC data (f) RandomForest for HASC data

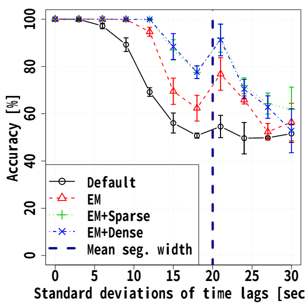
Figure 3: Mean errors for fixing segments (vertical) for various time lags (horizontal). The error bars $\pm 0.1\sigma$ are among different activity sequences.

In all the graphs, the errors increased with the size of the time lags. Among them, our proposed approaches of **EM+Dense** and

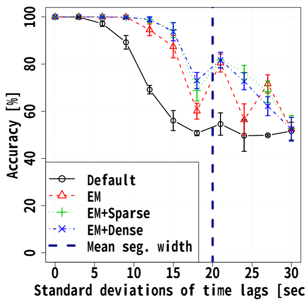
EM+Sparse outperformed the **Default** except for a few points, such as amending 20–30 s of lag standard deviations in (d), or the 0 s in (e) and (f).

4.3.2 **Question 2**: *Can the fixed segments improve the recognition accuracy?*

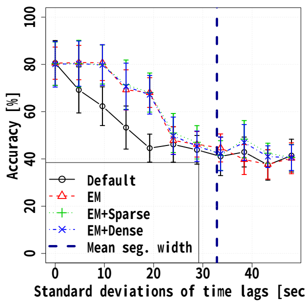
Figure 4 shows the experimental results for Question 2. The horizontal axis denotes the size of the standard deviations [seconds] in creating time lags for inaccurate segments, and the vertical axis denotes the recognition accuracy (%). Each graph shows the types of evaluation data and machine-learning algorithms for constructing $\mathbf{P}(y_t|x_t)$. In each graph, each line shows either **Default**, **EM** **EM+Sparse**, or **EM+Dense** methods with the bar of $\pm 1\sigma$ among different activity sequences.



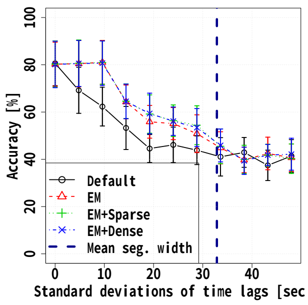
(a) NaiveBayes for simulation data



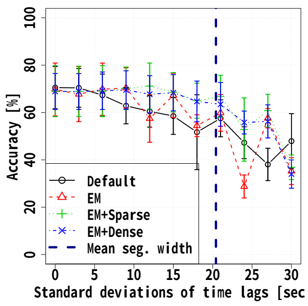
(b) RandomForest for simulation data



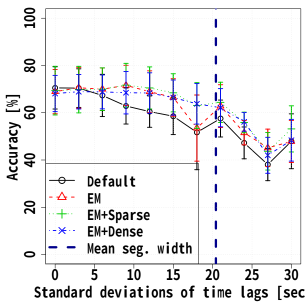
(c) NaiveBayes for HAR data



(d) RandomForest for HAR data



(e) NaiveBayes for HASC data



(f) RandomForest for HASC

Figure 4: Accuracies obtained through cross validation (vertical) for various time lags (horizontal). The error bars $\pm 1\sigma$ are among different activity sequences.

In all the graphs, the recognition accuracy decreases with an increase in the size of time lags. Among them, our proposed approaches of **EM**, **EM+Dense**, and **EM+Sparse** outperformed the **Default**. In most of the lags, **EM+Dense** and **EM+Sparse** were equally the best, followed by **EM**.

5. DISCUSSION

As aforementioned, for both Questions 1 and 2, the proposed approach outperformed the **Default** method for most of the time lags, and any of datasets and machine-learning algorithms.

The amount of improvement differed among the time lags; however, Figure 3 shows that at the time lags with half the length of the mean segment durations (00 [s] in Simulation, 16.44 [s] in HAR, 10.21 [s] in HASC), most of the graphs show that the improvements compared to the default methods are roughly more than constant.

These outperformances continue but become closer to those of the **Default** as the time lags increase. However, they are still superior in all the time lag ranges shown in the figure, except for some points in (d). The under performance in (d) is considered to be because of that the standard deviation of the time lag being closer to the segment width (dashed horizontal line); some segments were fixed incorrectly to the neighboring segments. Another approach would be required for these incorrect fixes.

As a result of such improvement, the recognition accuracies through cross validations were also improved, as shown in Figure 4. Moreover, although we did not compare the segmentation methods of **EM+Dense** and **EM+Sparse** with **EM** in the Figure 3, they show superior accuracy that of the **EM** under the mean segment duration lengths in Figure 4.

Interestingly, the accuracy at lag time 0 of the natural datasets in 4 are all the same in the **Default** and proposed approaches, while the proposed approaches were worse in Figure3 (c), (e), and (f). This implies that the proposed approach avoids overfitting even without cross validations.

On the basis of these results, our method could possibly determine the labels with wrong timestamps in the dataset. To demonstrate this, we checked through the original dataset, and found some inaccuracies. Figure 5 shows an example of labels modified to correct those from the HASC dataset. In both the graphs, the horizontal axis denotes the time and the black horizontal lines denote the labels. The upper graph shows the original labels, and the lower graph shows the labels modified using our method with **EM+Sparse**. The colored lines show the series of feature vectors.

As shown at approximately 200–450 s, the labels are modified and appear valid, following the feature vectors. Although the original labels should be as correct as possible, inaccuracy could occur as they are recorded manually. Our method can be utilized to fix such human errors.

6. RELATED WORK

Many studies on activity recognition based on supervised machine learning by using mobile sensors have been published, since its introduction in [2]. In supervised learning, the cost of generating training labels is problematic because labels are often generated through manual operations. Therefore, semisupervised machine learning was proposed and utilized in [13, 6].

Skitic et al. proposed activity recognition by utilizing training data without labels, and adding labels through semi-supervised learning based on self-training and co-training [15]. In [8], a function was proposed that projects a multi-dimensional space-specific feature value by using unlabeled and labeled data, where supervised learning uses a Support vector machine in the space of the projected label data. This enables the unlabeled data characteristics to affect the learning result. These studies are related as they use incomplete labels, but do not consider inaccurate time stamps.

In [14], *multi-instance learning* was used, that is, a machine learning method that can assume more than one sample for one label, and can train without knowing all the label data. This multi-

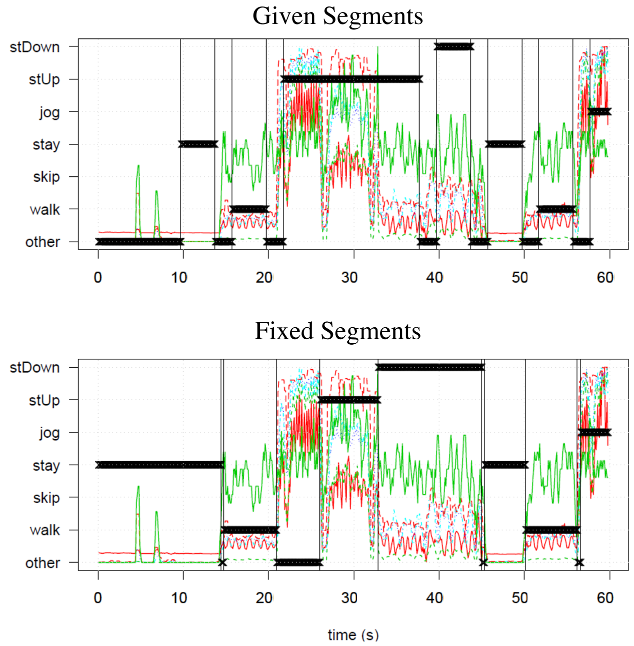


Figure 5: Example of labels modified to correct labels obtained from the HASC dataset. In both the graphs, the horizontal axis denotes time and the black horizontal lines denote the labels. The upper graph shows the original labels, and the lower graph shows the labels modified using our method with EM+Sparse. The colored lines show the series of feature vectors.

instance learning method was introduced in [17]. However, it was assumed that one or more true sample pairs are provided as the sample sets, and the study does not assume the segment inaccuracy, which leads to nonexistent true sample pairs but exists nearby.

In [11], instead of focusing on missing label times, a method to perform activity recognition only through activity order was proposed. In this method, the correct label is recognized through dynamic programming matching and supervised learning after segmentation and clustering. This method is effective when only the order is known. However, it is not able to attach a label to a specific time when the activity label time is inaccurate.

Multilabel learning is a type of machine learning that allows structured or multiple labels in the learning sample. The method was introduced in [4] and [17]. We focus on a multilabel learning in which the label may have a plurality of samples but only one true label exists. This is a special case of multilabel learning [9]. In [4], they solved the same problem as a convex programming problem of loss function, and applied it to videos for person labeling. We adopt the method by Jin et al. [9] as a building block of our approach in step **B)**; however, the method cannot be directly applied for inaccurate segments because we assume that a single label is provided for each time, and is solved in step **A)**. In addition, the result will be fragmented and not continuous; this is solved in step **C)**.

The method in [5] extends the technique of [9] by using conditional random fields often used in natural language processing. By doing so, even if many activity labels are given, machine learning can be performed. This is similar to our research in terms of extending the time-series data approach. However, they assumed that the application of more than one label to the data of one sample implies multiple persons. In contrast, we assume the occurrence of a time lag.

Toda et al. proposed the basic idea of utilizing multilabel learning for activity recognition [16]. However, they did not propose the segmentation method, which is important for fixing segments. In this paper, we introduce the newer idea of generating multilabels and segmentation, thus evaluating the accuracy and fixing segments thoroughly by using various datasets and machine-learning algorithms.

7. CONCLUSION

In this paper, we proposed an approach to improve mobile activity recognition, when a training data set with inaccurate segments is given, by converting the training dataset to multilabeled samples, training the dataset by using a multilabeled EM learning algorithm, and applying a segmentation method using the estimated labels and the original segment information. We evaluated the proposed approach for various datasets and machine-learning algorithms, and showed that the proposed approach outperforms the naive methods in terms of fixing the segments of the training data and improving the recognition accuracy through cross validation. Moreover, we demonstrated that we can determine inaccurate segments in the existing open dataset.

In the future, we intend to apply the approach for long-term data, such as ADL or nursing activity records in a hospital, where labels can be expected to be considerably inaccurate.

Acknowledgment

Part of this research is supported by the Grant-in-Aid for Scientific Research (B) “Sensor Context Estimation Techniques by Semantic and Physical Layer (Principal Investigator: Sozo Inoue).”

8. REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7657 LNCS, pages 216–223, 2012.

[2] Ling Bao and Stephen S. Intille. Activity Recognition from User-Annotated Acceleration Data, *Pervasive Computing,* 2004.

[3] Charles Bouveyron and Stéphane Girard. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658, 2009.

[4] Timothee Cour and Benjamin Sapp. Learning from Partial Labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

[5] Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. Sequence Learning from Data with Multiple Labels. (Figure 1).

[6] Yves Grandvalet and Y Bengio. Semi-supervised learning by entropy minimization. 2005.

[7] I Guyon and A Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] Tâm Huáżşnh and Berat Schiele. Towards less supervision in activity recognition from wearable sensors. In *Proceedings - International Symposium on Wearable Computers, ISWC*, pages 3–10, 2007.

[9] R Jin and Z Ghahramani. Learning with multiple labels. *Advances in neural information ...*, pages 897—-904, 2002.

[10] Nobuo Kawaguchi, Nobuhiro Ogawa, and Yohei Iwasaki. Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings. In *Proceedings of the 2nd Augmented Human International Conference*, page 27, 2011.

[11] Kazuya Murao and Tsutomu Terada. Labeling method for acceleration data using an execution sequence of activities. *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct*, pages 611–622, 2013.

[12] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with Noisy Labels. *Advances in neural information processing systems*, pages 1196–1204, 2013.

[13] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2521–2537, 2011.

[14] Maja Stikic and Bernt Schiele. Activity recognition from sparsely labeled data using multi-instance learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5561 LNCS, pages 156–173, 2009.

[15] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 81–88, 2008.

[16] Takamichi Toda, Sozo Inoue, Shota Tanaka, and Naonori Ueda. Training Human Activity Recognition for Labels with Inaccurate Time Stamps. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 863–872, 2014.

[17] Zhi Hua Zhou, Min Ling Zhang, Sheng Jun Huang, and Yu Feng Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176:2291–2320, 2012.