

Semi-Markov conditional random fields for accelerometer-based activity recognition

La The Vinh · Sungyoung Lee · Hung Xuan Le ·
Hung Quoc Ngo · Hyoung Il Kim · Manhyung Han ·
Young-Koo Lee

Published online: 12 March 2010
© Springer Science+Business Media, LLC 2010

Abstract Activity recognition is becoming an important research area, and finding its way to many application domains ranging from daily life services to industrial zones. Sensing hardware and learning algorithms are two important components in activity recognition. For sensing devices, we prefer to use accelerometers due to low cost and low power requirement. For learning algorithms, we propose a novel implementation of the semi-Markov Conditional Random Fields (semi-CRF) introduced by Sarawagi and Cohen. Our implementation not only outperforms the original method in terms of computation complexity (at least 10 times faster in our experiments) but also is able to capture the interdependency among labels, which was not possible in the previously proposed model. Our results indicate that the proposed approach works well even for complicated activities like eating and driving a car. The average precision and recall are 88.47% and 86.68%, respectively, which are higher than results obtained by using other methods such as Hidden Markov Model (HMM) or Topic Model (TM).

Keywords Activity recognition · Wearable sensors · Accelerometer · Hidden Markov Model (HMM) · Conditional Random Fields (CRF)

1 Introduction

Nowadays, activity recognition is an increasingly important research area. The modern life style tends to involve in more

sedentary jobs, while there are growing evidences in the relationship between common health problems such as diabetes, cardiovascular, osteoporosis and the level of physical activity [24]. Therefore, simple monitoring will not protect anybody from any disease but may help to assess and then alter the life style, which in turn could result in health benefits. In addition, activity recognition has been considered to be a potential factor in improving convenience as well as productivity at the work place; for example, in smart hospitals [6, 18], in aircraft maintenance [10], or in a workshop [13]. Also, such activity recognition systems can be used to predict abnormal behaviors such as falling down [14] for emergency response in health-care systems.

There are various approaches using video [4] and deployed sensors [21], many researchers, however, have used accelerometers in their research work due to their low cost, low power requirement, portability, and versatility characteristics. Therefore, we also use accelerometer-based input for our activity recognition system. With respect to recognition methods, sliding window approach is commonly used in accelerometer-based activity recognition [1, 13, 17, 20]. However, in most cases, a sliding window cannot cover one complete activity, since the duration of different activities usually varies significantly and the start time of an activity in a continuous stream is unknown in advance. Thus, the sliding window approach may produce fragments of activities making it difficult to obtain comprehensive models to satisfy the performance requirements of a continuous activity recognition system. One feasible solution for this problem is to take into account the duration of activities so that some short-length fragments can be eliminated. In addition to the problem associated with sliding windows, the interdependency among activities is another issue, which should be considered when detecting activities in a continuous data stream.

L.T. Vinh · S. Lee (✉) · H.X. Le · H.Q. Ngo · H.I. Kim · M. Han ·
Y.-K. Lee
Dept. of Computer Engineering, Kyung Hee University, 446-701
Room 351, Seocheon-dong, Giheung-gu, Yongin-si,
Gyeonggi-do, Republic of Korea
e-mail: sylee@oslab.khu.ac.kr

Nevertheless, to the best of our knowledge, none of the existing activity recognition models is able to handle all the afore mentioned problems, especially, for large-scale activity recognition systems. Therefore, in this study, we propose a novel method for activity recognition, which is based on the work from Sarawagi and Cohen [19], to model the duration as well as the interdependency of activities. Furthermore, we introduce our clever caching algorithm to overcome the high computation complexity of the original work [19].

2 Paper contribution and outline

Our contributions in this work are three folds. First, we propose a novel implementation of semi-Markov Conditional Random Fields (semi-CRF) that is superior to one proposed in [19]. Second, we propose an efficient algorithm for parameter estimation, which runs much faster than the original training algorithm in [19]. Third, we apply the proposed semi-CRF to accelerometer-based activity recognition with a large-scale dataset.

The rest of this paper is organized as follows. We briefly survey related work and their results in Sect. 3. Section 4 describes the background of our work which includes the standard CRF [9, 23] and an existing implementation of semi-CRF [19]. We also point out limitations of these models, which are solved by our proposed approach as described in Sect. 5. Section 6 shows how the proposed semi-CRF model can be applied to accelerometer-based activity recognition. In Sect. 7, we discuss in detail our experiments and results to show our improvements. We conclude the paper and outline future work in Sect. 8. Finally, we present details about our algorithms in the appendix section.

3 Related work

So far, many algorithms have been proposed for accelerometer-based activity recognition. Decision tree, support vector machine and some other kinds of classification methods were evaluated in [1, 17]. To make use of the sequential structure of activities, Hidden Markov Model (HMM) was used in [20]. Recently, Conditional Random Fields model (CRF) was introduced as a much better approach compared to HMM in sequential modeling [9]. Thus, some researchers have successfully applied CRF to activity recognition [12, 23].

A limitation of both conventional HMM and first-order CRF is the Markovian property, which assumes that the current state depends only on the previous state. Because of this assumption, the labels of two adjacent states must be supposed to occur successively in the observation sequence.

Unfortunately, the presumption is not always satisfied in reality. For example, in the activity recognition problem, two expected activities (activities that we want to recognize) are often separated by irrelevant activities (activities that we do not intend to detect). Furthermore, constant self-transition probabilities cause the distribution of state's duration to be geometric [16] which is inappropriate to the real activity duration model.

In [19], Sarawagi and Cohen have shown that semi-CRF is capable of using an explicit duration model. It, however, increases the computation complexity of forward and backward algorithms by D times from $O(TM^2)$ to $O(TM^2D)$, where T, M, D are the length of the input sequence, the number of possible label values, and the maximum duration length, respectively. In [5], the proposed semi-CRF model requires a complexity of $O(TM^2D)$ for estimating each gradient. If we have N parameters to be optimized, the computation load will be $O(NTM^2D)$, which is very high. Truyen et al. [22] introduced a more complicated model, called Hierarchical Semi-Markov Conditional Random Fields (HSCRF) and demonstrated that HSCRF could be converted to semi-CRF as a special case. Nevertheless, their conversion did not show any improvement in the complexity required for the optimization of the model's parameters. In [15], the authors proposed a method to decrease the computational cost of semi-CRF. They, however, utilized a Bayes filter to eliminate some sequences from the computation. The approach, therefore, did not keep the originality of the problems. In short, semi-CRF model is a potential solution for modeling sequential data like activity data. However, the current training algorithms for semi-CRF require high complexity, making the model impractical in large-scale systems. Furthermore, the semi-CRF model, proposed in [19], is still not able to solve the long-range transition difficulty. The authors in [11] used a high order CRF model to capture long-range transitions in case of predefined transition order. However, if the order of transition is not known exactly, their method cannot be used. It also should be noted that not many research contributions are made in scalable activity recognition [7].

In this study, we propose to overcome the above limitations of the existing work by introducing our novel semi-CRF to model both the duration and the interdependency of activities. Additionally, we develop a fast training algorithm for our model making it suitable for scalable activity recognition applications.

4 Background

In this section, we briefly review the theory of Conditional Random Fields [9] and its extension [19]. We also point out the limitations of the existing work at the end of this section.

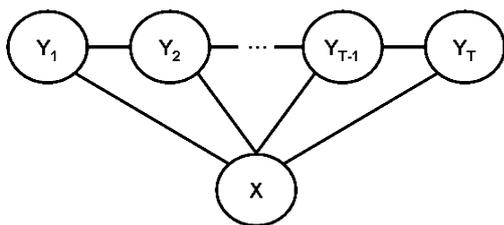


Fig. 1 The graphical structure of a linear-chain Conditional Random Fields model

4.1 Conditional random fields

In conventional CRF [9], the input sequence X and the corresponding label sequence Y of length T are given in the form

$$X = \{x_1, x_2, \dots, x_T\}, \tag{1}$$

$$Y = \{y_1, y_2, \dots, y_T\}. \tag{2}$$

In this work, we consider only linear chain CRF as depicted in Fig. 1. The likelihood of the labeled data is calculated as

$$P(Y|X) = \frac{\prod_{t=1}^T \Psi(y_{t-1}, y_t, X)}{Z_X}, \tag{3}$$

$$\Psi(y_{t-1}, y_t, X) = e^{W^T F(y_{t-1}, y_t, X)}, \tag{4}$$

$$Z_X = \sum_{Y'} \prod_{t=1}^T \Psi(y'_{t-1}, y'_t, X), \tag{5}$$

where F is a column vector of feature functions, W is a column vector of model parameters, Ψ is called potential function. Z_X , the normalization factor, is computed by using forward/backward variables

$$\alpha_t(y_t) = \sum_{y_{t-1}} \Psi(y_{t-1}, y_t, X) \alpha_{t-1}(y_{t-1}), \tag{6}$$

$$Z_X = \sum_{y_T} \alpha_T(y_T). \tag{7}$$

Model parameters are estimated so that $P(Y|X)$ is maximized, or equivalently, $L(Y|X) = \log(P(Y|X))$ is maximized. Since, solving $\frac{dL}{dw_i} = 0, i = 1, 2, 3, \dots, N$, is intractable, numerical methods such as stochastic gradient ascent method are applied to optimize the concave function $L(Y|X)$.

4.2 Semi-Markov conditional random fields

Since the conventional CRF is limited to the Markovian assumption that the label y_t at time t depends only on the previous label y_{t-1} , it is not able to capture the duration distribution as well as the interdependency of labels. Therefore, semi-Markov model is proposed to handle these issues. In [19], Sarawagi and Cohen described a method for

learning and inferring with semi-CRF. The authors include in each state a label, a beginning time and an ending time. Thus, a new state is defined as

$$s_i = (y, b, e) \quad i = 1, 2, \dots, P, \tag{8}$$

where P is the length of the sequence $S = s_1, \dots, s_P$, which is constructed from input labels $Y = (y_1, y_2, \dots, y_T)$. y, b , and e are label, beginning time, and ending time of the state s_i , respectively. For example, if we have $Y = (1, 1, 2, 2, 2, 3, 4, 4)$ then $S = \{(1, 1, 2), (2, 3, 5), (3, 6, 6), (4, 7, 8)\}$. Originally, the beginning and ending time must satisfy the following constraints.

$$s_i.b \leq s_i.e \quad i = 1, 2, \dots, P, \tag{9}$$

$$s_i.e + 1 = s_{i+1}.b \quad i = 1, 2, \dots, P - 1, \tag{10}$$

$$s_1.b = 1, \tag{11}$$

$$s_P.e = T. \tag{12}$$

Now, instead of computing the likelihood of Y given X , the likelihood of S given X is estimated by

$$P(S|X) = \frac{\prod_{i=1}^P \Psi(s_{i-1}, s_i, X)}{Z_X}, \tag{13}$$

$$Z_X = \sum_{S'} \prod_{i=1}^{P'} \Psi(s'_{i-1}, s'_i, X), \tag{14}$$

where $\Psi(s_{i-1}, s_i, X)$ encodes the potential of the transition from s_{i-1} to s_i . In the following equations, we suppose that any function $\Psi(s_{i-1}, s_i, X)$ can be rewritten in the form $\Psi(s_{i-1}.y, s_i.y, X, s_i.b, s_i.e)$. For example, with the sequence S above, we can rewrite $\Psi(s_1, s_2, X)$ as $\Psi(1, 2, X, 3, 5)$.

$$\Psi(s_{i-1}, s_i, X) = e^{W^T F(s_{i-1}, s_i, X)}, \tag{15}$$

where $W = [w_1, w_2, \dots, w_N]^T$ is a column vector of model parameters,

$$F(s_{i-1}, s_i, X) = \begin{bmatrix} f^1(s_{i-1}, s_i, X) \\ f^2(s_{i-1}, s_i, X) \\ \dots \\ f^N(s_{i-1}, s_i, X) \end{bmatrix}$$

is a column vector of feature functions. In (13) and (14), we can consider the product of potential functions Ψ over all transitions of a sequence as the potential of the sequence. Thus, we can rewrite (13) in the form

$$P(S|X) = \frac{\text{Pol}(S)}{\sum_{S'} \text{Pol}(S')}, \tag{16}$$

where

$$\text{Pol}(S) = \prod_{i=1}^P \Psi(s_{i-1}, s_i, X) \tag{17}$$

is the potential of the sequence $S = s_1, s_2, \dots, s_P$. The forward algorithm and parameter estimation are implemented based on the following equations [19]

$$\alpha(t, y) = \sum_{d=1}^D \sum_{y'} \alpha(t-d, y') \Psi(y', y, X, t-d, t) \tag{18}$$

$t = 1, 2, \dots, T,$

$$Z_X = \sum_y \alpha(T, y), \tag{19}$$

$$\frac{dZ_X}{dw_k} = \sum_y \eta^k(T, y), \tag{20}$$

$$\begin{aligned} \eta^k(t, y) = & \sum_{d=1}^D \sum_{y'} ((\eta^k(t-d, y') \\ & + \alpha(t-d, y') f^k(y', y, X, t-d, t)) \\ & \times \Psi(y', y, X, t-d, t)), \end{aligned} \tag{21}$$

for $t = 1, 2, \dots, T$ and $k = 1, 2, \dots, N$. Where N is the number of model’s parameters, D is the maximum duration of a label.

Based on equations from (13) to (21), the derivative of log likelihood of S given X is calculated as

$$\begin{aligned} \frac{d}{dw_k} \log(P(S|X)) = & \sum_{i=1}^P f^k(s_{i-1}, s_i, X) \\ & - \frac{\sum_y \eta^k(T, y)}{\sum_y \alpha(T, y)}. \end{aligned} \tag{22}$$

Although, the semi-CRF model proposed in [19] is able to utilize an explicit duration model, it is still not able to capture long-range transitions among labels because of the equality in (10). For example, given a sequence of activity labels $Y = \{\text{Eating, Eating, IA, IA, Cleaning, Cleaning, IA, IA, IA}\}$, where “IA” stands for “Irrelevant Activity”. Then, the semi-Markov sequence is $S = \{(\text{Eating}, 1, 2), (\text{IA}, 3, 4), (\text{Cleaning}, 5, 6), (\text{IA}, 7, 9)\}$. Clearly, herein, we can utilize the transitions from “Eating” to “IA” or from “IA” to “Cleaning”, but we are not able to take advantage of the transition from “Eating” to “Cleaning”. Hereafter, in our equations we will use symbols y and “IA” to represent labels of expected and irrelevant activities, respectively. In the next section, we tackle the difficulty by relaxing the constraints in (10), (11), and (12) so that our model can skip irrelevant activities and directly model the transitions between two expected activities.

5 Semi-CRF with discontinuous states

In this section, we present details of our proposed model, which is applied to accelerometer-based activity recognition, based on the semi-CRF model introduced in [19].

To handle the problem of contiguous state labels, we use inequalities instead of the equalities in (10), (11), and (12), so we have $0 < s_i.b \leq s_i.e < s_{i+1}.b \leq s_{i+1}.e \leq T$ $i = 1, 2, \dots, P - 1$. For example, given a sequence of activity labels $Y = \{\text{Eating, Eating, IA, IA, Cleaning, Cleaning, IA, IA, IA}\}$, the corresponding semi-Markov sequence is $S^9 = \{(\text{Eating}, 1, 2), (\text{Cleaning}, 5, 6)\}$. The labels, located in the time slots which are not occupied by any expected activities, are “IA” by default. The superscript ⁹ represents the length of the original sequence Y of S^9 . Obviously, by this way we are able to utilize the transition from “Eating” to “Cleaning” directly although they are separated by “IA”.

In our approach, we suppose that the duration potential, the transition potential, and the observation potential are independent of each other. Thus, we can rewrite the potential function as below

$$\Psi(s_{i-1}, s_i, X) = \begin{pmatrix} e^{Q^{Tr}(s_{i-1}, s_i, X)} \times \\ e^{Q^D(s_{i-1}, s_i, X)} \times \\ e^{Q^O(s_{i-1}, s_i, X)} \end{pmatrix}. \tag{23}$$

The weighted transition potential function is given by

$$\begin{aligned} Q^{Tr}(s_{i-1}, s_i, X) \\ = \sum_{y', y} w^{Tr}(y', y) \delta(s_{i-1}.y = y', s_i.y = y), \end{aligned} \tag{24}$$

where $w^{Tr}(y', y)$ is the weight of transition from y' to y and δ is given by

$$\delta(A) = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{if } A \text{ is false.} \end{cases} \tag{25}$$

The weighted duration potential function of an expected activity is calculated as

$$\begin{aligned} Q^D(s_{i-1}, s_i, X) \\ = \sum_{y, d} G^D(y, d) \delta(s_i.y = y, d = s_i.e - s_i.b + 1) \\ = \sum_{y, d} w^D(y) \frac{(d - m_y)^2}{2\sigma_y^2} \delta(s_i.y = y, d = s_i.e - s_i.b + 1), \end{aligned} \tag{26}$$

where $w^D(y)$ is the duration weight of y . m_y and σ_y are the empirical average and standard deviation of state y ’s duration, respectively, which can be easily extracted from training data. Figure 2 depicts the shape of the corresponding potential function $e^{G^D(y, d)}$ with three different values of y .

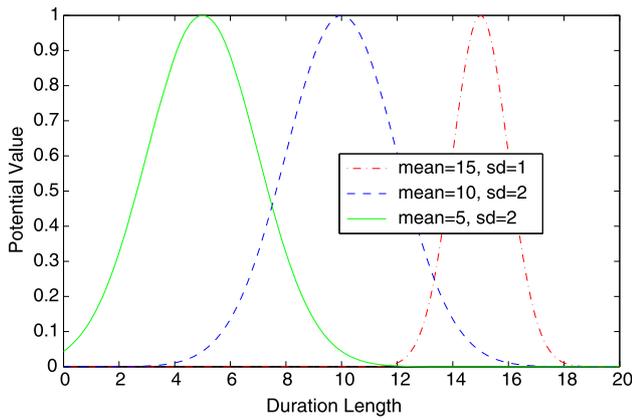


Fig. 2 (Color online) Duration potential with different values of mean and standard deviation. If a detected segment has a length of, for example 6, then it most likely belongs to the same class with the label, whose duration potential is presented in the green

Clearly, the most likely duration (at the center of the bell) has the highest potential value. Since irrelevant activities can have an arbitrary length, we do not model the duration of such activities because modeling will increase the possible maximum duration length, D , which may result in high complexity.

Note that in (26) we assume that the duration of an activity has a Gaussian-like distribution. Although the assumption is not always true, it is reasonable to assume that since most activities often last around a constant amount of time.

Next, we define the weighted observation potential function as

$$\begin{aligned}
 Q^O(s_{i-1}, s_i, X) &= \sum_{y, t_1, t_2} (G_y(y, t_1, t_2) \delta(s_i.y = y, s_i.b = t_1, s_i.e = t_2) \\
 &\quad + G_{IA}(IA, t_1, t_2) \delta(s_{i-1}.e + 1 = t_1, s_i.b - 1 = t_2)), \tag{27}
 \end{aligned}$$

where

$$G_y(y, t_1, t_2) = \sum_{t=1}^{t_2} \sum_o w^O(y, o) \delta(x_t = o), \tag{28}$$

$$G_{IA}(IA, t_1, t_2) = \sum_{t=1}^{t_2} \sum_o w^O(IA, o) \delta(x_t = o), \tag{29}$$

where $w^O(y, o)$ and $w^O(IA, o)$ in that order are the weights of the observation given that input symbol o is observed in state with label y and IA . For convenience in the presentation of the following equations, we denote $G(y, t_1, t_2) = G_y(y, t_1, t_2) + G^D(y, t_2 - t_1 + 1)$ as a combined potential function.

Our approach is similar to that in [19], but we allow discontinuity in the time of state by using $s_{i+1}.b > s_i.e$ instead

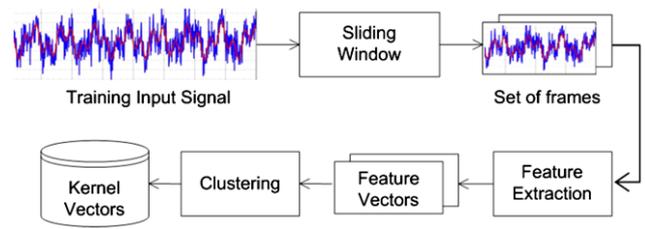


Fig. 3 Training Data Clustering

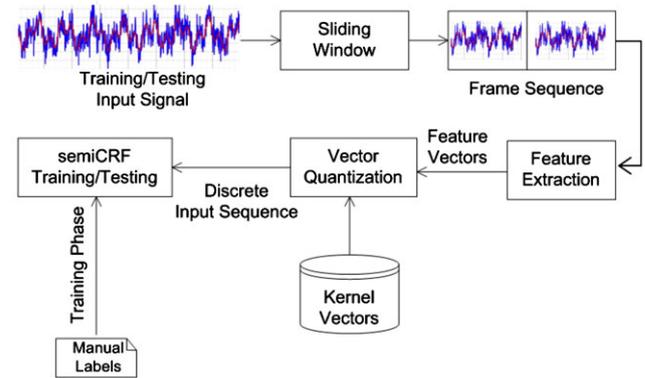


Fig. 4 Training/Testing semi-CRF with discrete input sequences

of $s_{i+1}.b = s_i.e + 1$ as in [19]. The inequality enables our model to skip irrelevant activities and to directly model the transition between two expected activities. For the training and inference algorithms, we propose novel implementations of the forward, backward algorithms, gradient calculating algorithm and Viterbi algorithm. Details of our algorithms are presented in the appendix section at the end of this paper.

6 The proposed activity recognition framework

In this section, we present how the proposed semi-CRF model can be applied to accelerometer-based activity recognition.

In our system, discrete values were used as the input of the semi-CRF. Therefore, we first quantized the continuous input signal from the accelerometers using Linde-Buzo-Gray (LBG) algorithm. Figure 3 illustrates a block diagram of the clustering module. In this module, we employed overlapped sliding windows to chop the signal into equal-length frames. After windowing, feature vectors were extracted from signal frames and fed into a LBG clustering function to construct a codebook. Then, every input feature vector was quantized to an integer value, which was the index of the nearest codebook vector. The output sequence of discrete values was used for training or testing the semi-CRF model, as it is shown in Fig. 4. We conducted experiments to analyze the effect of different parameter's values on the achieved results. The percentage of the overlap between two contiguous windows, the length of each

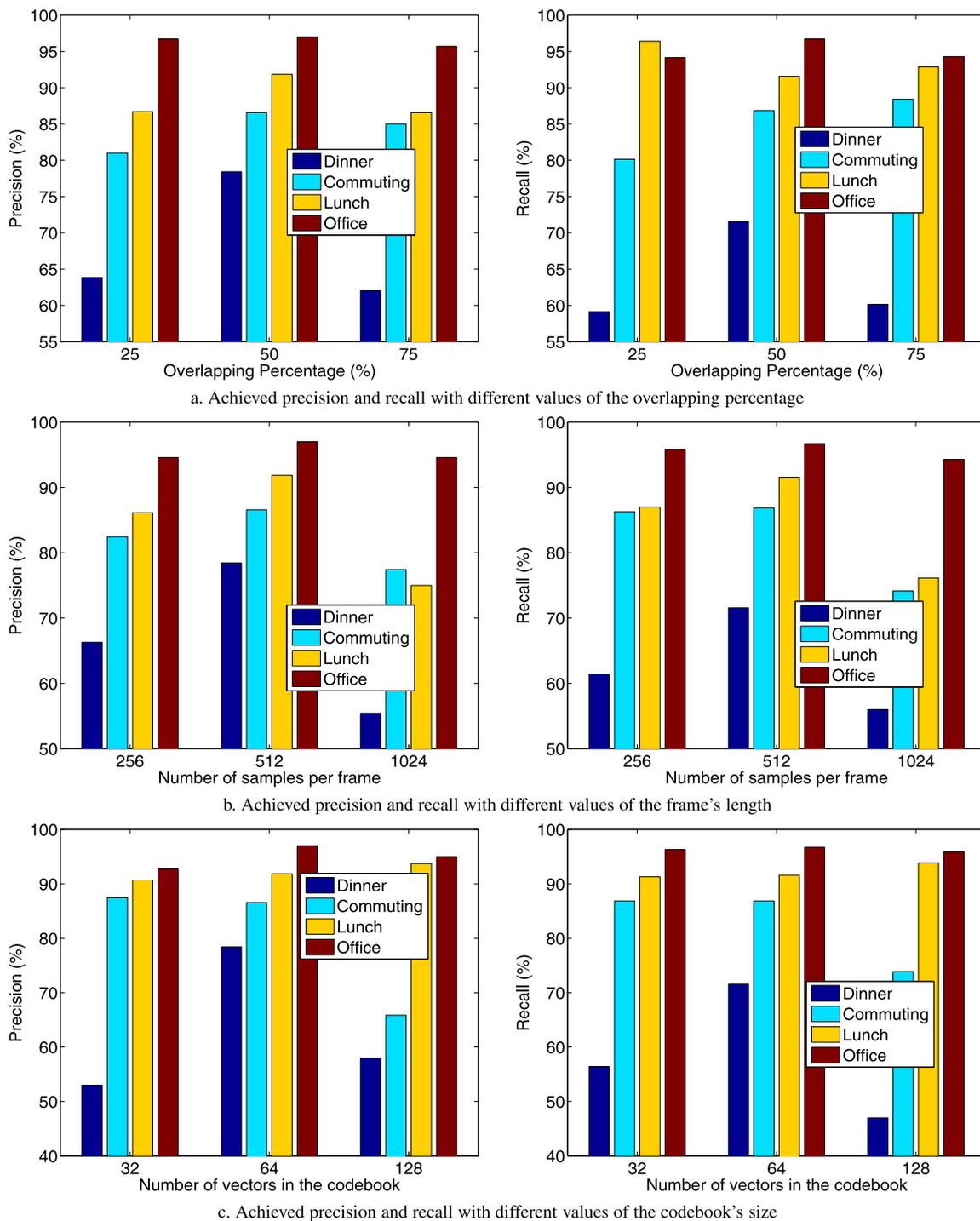


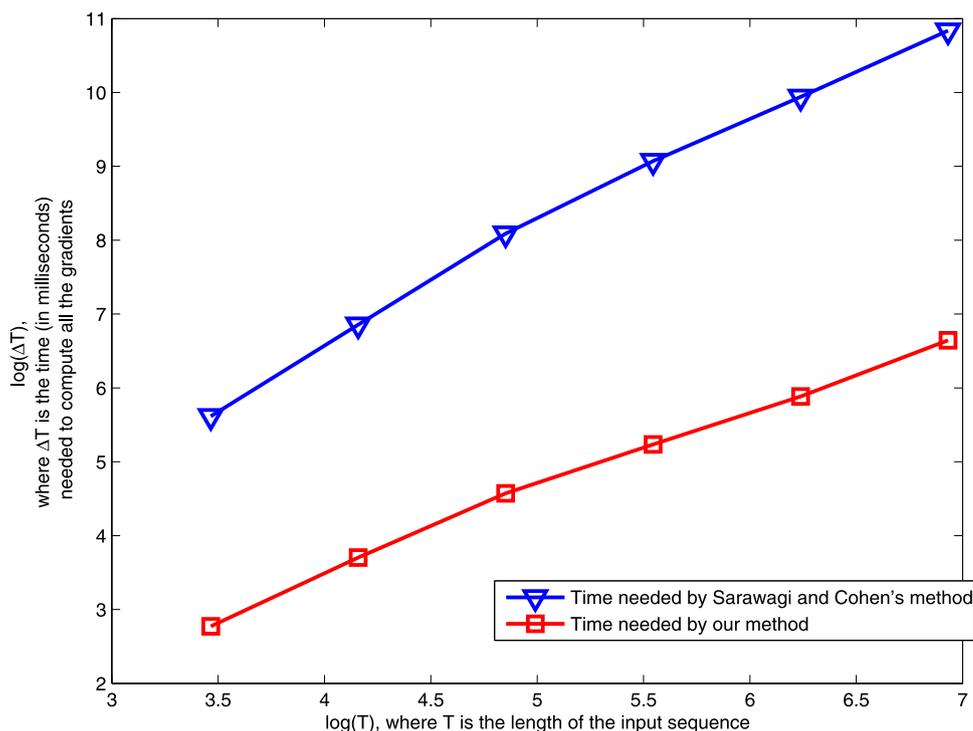
Fig. 5 Achieved precision and recall with different parameter's values

window, and the codebook's size were chosen based on our experiments. Figure 5 shows the results obtained with different values of these parameters. From the results depicted in Fig. 5, we decided that the overlap portion was 50%, the window's length was 512 samples, and the number of codebook vectors was 64.

7 Evaluation

In this section, we first show that our proposed algorithm achieves a remarkable improvement in terms of computation complexity compared to the previous work. Then, we present the recognition results with an available large-scale

Fig. 6 Average time needed for computing all the gradients. Herein, the number of labels (M) is 4, the maximum duration (D) is 16, the codebook's size (V) is 128, the length of the input sequence (T) changes from 32 to 1024. Therefore, the number of gradients is $M + M^2 + MV = 532$



dataset. From the achieved results, we show that the proposed method not only speeds up the calculation significantly, but also produces much better accuracy in recognition. All the experiments were implemented in C++ on a computer with Intel dual core 1.83 GHz processor and 512 MB RAM.

7.1 Complexity analysis

In the algorithm proposed in [19], the estimation of (22) requires that $\alpha(t, y)$ and $\eta^k(t, y)$ are pre-calculated for all possible values of t and y . Each of them needs a complexity of $O(MD)$ as can be seen in (18) and (21). Therefore, the complexity per gradient of (22) is proportional to $O(TM^2D)$. Hence, the estimation of gradients for all N model's parameters takes $O(NTM^2D)$.

In our solution, gradients are computed by using (52), (60), (62), and (63). It is obvious that if $\alpha, \gamma, \lambda, \beta, \theta, \zeta,$ and ν are cached, then estimating the above equations requires a maximum complexity of $O(TD)$. Hence, for optimizing N parameters, our algorithm requires only $O(NTD)$ to completely calculate all gradients. Nevertheless, we need to take into account the extra time of estimating the cached variables. As shown in the pseudo-code for the forward and backward algorithm, $\alpha, \gamma, \lambda, \beta, \eta,$ and ζ can be computed with $O(2TM(M + D))$. Meanwhile, from (59) and (68) we see that θ and ν take $O(TMD)$ and $O(TM^2)$, respectively. Totally caching these variables requires a complexity of $O(3TM(M + D))$.

It can be seen that our improvement completely comes from the caching mechanism. In our gradient (52), (60), (62) and (63) we utilize different partitioning methods, which take into account the characteristics of the gradients. For example, in (52) we partition the set of all length- T label sequences (S^T) into subsets $(\cup_t A_t^{y'} \oplus \Omega_{t+1}^y)$ based on the occurrence time of a transition from y' to y . Meanwhile in [19] a fixed partitioning method, where S^T was always divided into subsets based on the length of the last segment, was used regardless of different characteristics of the gradients. This is the reason why our algorithm achieves much more efficient caching results.

Herein, we take a numerical example to compare $O(NTM^2D)$, which is the estimated complexity in [19], to $O(3TM(M + D)) + O(NTD)$, our algorithm complexity. Suppose that we need to compute $N = 1000$ gradients of an input sequence, which has the length $T = 1000$, the maximum duration $D = 100$, the number of labels $M = 8$, then the former is about 64×10^9 , the latter is about 10^9 . In addition, Fig. 6 illustrates another comparison of the two complexities with $N = 532, M = 4, D = 16$ and T changes from 32 to 1024. Both algorithms were evaluated on the same computer with the same dataset. The amount of time which was required by the method proposed by Sarawagi and Cohen in [19] is marked in blue, time consumed by our algorithm is marked in red. Obviously, there is a remarkable improvement in our complexity since our time requirement is at least 10 times less than the computation time required by Sarawagi and Cohen's method.

Table 1 Low level activities occurring during the routines [8]

Activity	Average duration	Occurrences	Total
Sitting/desk activities	49.41 min	54	3016.0 min
Unlabeled	1.35 min	239	931.3 min
Having dinner	17.62 min	6	125.3 min
Walking freely	2.86 min	38	124.2 min
Driving car	10.37 min	10	120.3 min
Having lunch	10.95 min	7	75.1 min
Discussing at white board	12.80 min	5	62.7 min
Attending a presentation	48.9 min	1	48.9 min
Driving a bike	11.82 min	4	46.3 min
Walking while carrying something	1.43 min	10	23.1 min
Walking	2.71 min	7	23.0 min
Picking up mensa food	3.30 min	7	22.6 min
Sitting/having a coffee	5.56 min	4	21.8 min
Queuing in a line	2.89 min	7	19.8 min
Using the toilet	1.95 min	2	16.7 min
Washing dishes	3.37 min	3	12.8 min
Standing/having a coffee	6.7 min	1	6.7 min
Preparing food	4.6 min	1	4.6 min
Washing hands	0.32 min	3	2.2 min
Running	1.0 min	1	1.0 min
Wiping the whiteboard	0.8 min	1	0.8 min

7.2 Experiments

In our experiments, we used a dataset of long-term activities, which has been published at <http://www.mis.informatik.tu-darmstadt.de/data>. The dataset contains 7 days of continuous data (except the sleeping time), measured by two tri-axial accelerometers, one on the wrist and the other in the right pocket. The sensor's sampling frequency was set to 100 Hz. In the published dataset, the author calculated the mean value of every 0.4 s window, so the actual sampling frequency was about 2.5 Hz. In total, it has 34 labeled activities, of which a subset of 24 activities occurred during the routines. Table 1 lists all the annotated activities, which were grouped by the authors in [8] into 5 daily routines as seen in Table 2. To compare our method with the existing methods, which were evaluated in [8] and [2], we kept all the data settings unchanged.

From our experiment's results illustrated in Fig. 5, we chose 50%-overlapped-sliding windows which had a length of 512 samples (about 3.41 minutes). Within a window, mean, standard deviation, and mean crossing rate were extracted from each signal. Then, these values were combined with the time of frame to form a feature vector. Finally, we followed leave-one-out cross validation rule to measure the results of recognition as can be seen in Table 3. Figure 7

Table 2 Daily routines

Routine	Occurrences
Dinner	7
Commuting	14
Lunch	7
Office work	14
Unknown (null)	>50

demonstrates an example of recognized routines together with the ground truth.

Our results show a considerable improvement compared to [2, 8], except dinner routine which has lower precision and recall than those of [2], with 3 other routines we obtain better results. Since we utilize similar features (mean, standard deviation, time) as in the original work, the improvement can be explained as the result of taking into account the interdependency together with the duration of activities.

Nevertheless, when trying to explain the results obtained with the dinner routine, we find out that our precision and recall are affected by the fragmentation of the routine. As seen in Table 4, the worst result, decreasing the overall achievement, is achieved with dinner routine of day 2, which is interrupted by other activities such as walking, and carrying something. Meanwhile, we still obtain quite good results (on

Table 3 Recognition result: the first and second column contain the results which are evaluated by Huynh et al. in [8] using a HMM, and a TM, respectively, the third column contains the achievement of Ulf

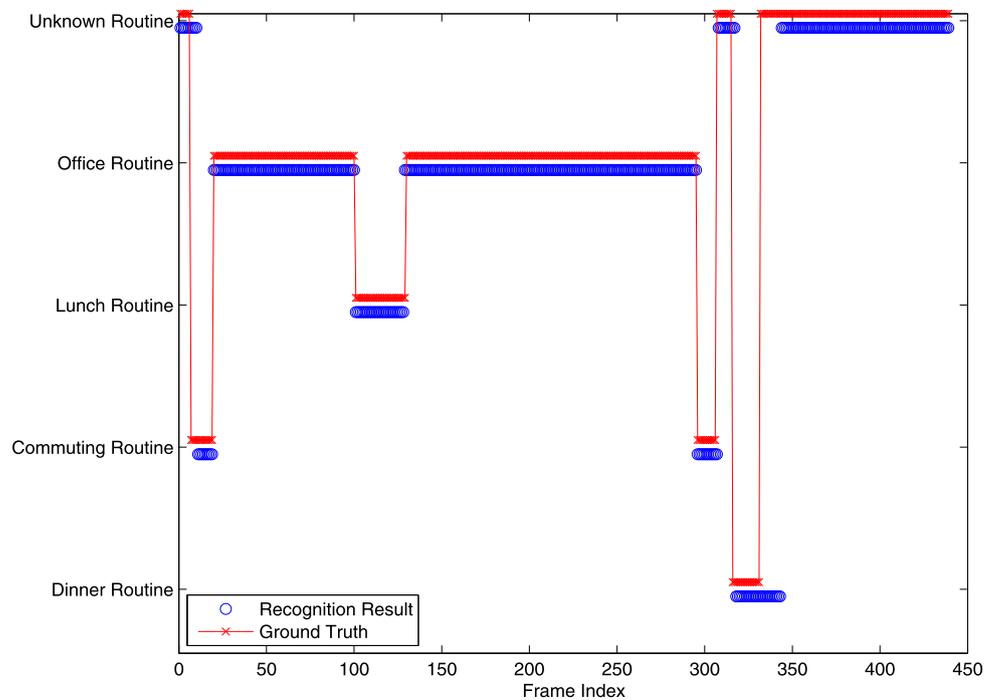
Blanke and Bernt Chiele using boosting techniques [2], our results are presented in the last column. All the recognition experiments were evaluated with the same dataset

	Baseline (HMM)	Huynh et al.	Ulf Blanke et al.	Our method
Routines	Precision/Recall (%)	Precision/Recall (%)	Precision/Recall (%)	Precision/Recall (%)
Dinner	88.60/27.30	56.90/40.20	85.27/ 90.48	78.43/71.57
Commuting	72.60/31.50	83.50/71.10	81.77/82.36	86.57/86.86
Lunch	84.40/80.70	73.80/70.20	84.56/90.04	91.86/91.57
Office	89.20/91.10	93.40/81.80	98.12/93.63	97.00/ 96.71

Table 4 Sequence of activities, which occur in the dinner routine. As can be seen, a dinner routine often contains some related activities such as having dinner, sitting, and washing dishes. However, in day 2 the dinner routine is interrupted by walking and carrying something

Day	Sequence of activities	Precision (%)	Recall (%)
1	Having dinner, sitting	100	100
2	Having dinner, walking, sitting, walking, sitting, carrying something, walking, washing dishes	0	0
3	Having dinner, sitting, washing dishes	60	100
4	Having dinner	100	76
5	Having dinner, washing dishes	95	95
6	Having dinner, sitting	100	45
7	Having dinner, sitting, washing dishes	94	85

Fig. 7 A single day recognized routines



average 91.00% and 90.34% for precision and recall, respectively) with other days, which are not fragmented.

Table 5 shows an example of transition matrix after training. It can be seen that the potential of most likely transitions such as “office work–lunch”, “commuting–dinner” or

“lunch–office” is always the highest to encourage the prediction of these events.

In addition to the accuracy, our solution also has a practical training time. With 72 hours of training data, our algorithm takes about 2 hours to complete parameter estimation

Table 5 Transition weights

	Dinner	Commuting	Lunch	Office
Dinner	-9.67	-10.99	-9.34	-8.99
Commuting	-2.50	-8.38	-6.80	-4.20
Lunch	-7.08	-9.39	-8.23	-6.12
Office	-9.60	-9.38	-7.51	-8.38

in a system with Intel dual core 1.83 GHz processor and 512 MB RAM.

In conclusion, our proposed model and algorithm work well in practice, making the semi-CRF model practical for large-scale activity recognition.

8 Conclusion

In this paper, we have presented a novel implementation of semi-Markov Conditional Random Fields and fast algorithms for gradient calculation. The solution not only is able to make use of the interdependency and the duration of activities to increase the accuracy, but also takes a practical amount of time for parameter estimation. Although the algorithm produced a low accuracy in some particular case, overall, we have shown that our approach obtains better results when compared to others. In this study, we only used activity recognition as our target application. However, the semi-CRF model can be extensively applied to other fields such as natural language processing or gene prediction. Our work, therefore, may bring a significant contribution not only to activity recognition, but also to a broader range of research areas.

For the future work, we plan to extend the proposed algorithms to handle continuous inputs and disparate sensors, such as audio sensors, gyroscope sensors, and video sensors. It would require further research to find out the mechanism of their correlation and the role of each sensor in activity recognition.

Acknowledgements This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0002)). This work was also sponsored by the IT R&D program of MKE/KEIT, [10032105, Development of Realistic Multiverse Game Engine Technology], and by the Basic Science Research Program funded by National Research Foundation (2009-0076798). We also would like to acknowledge Huynh and his colleges for the publication of the activity dataset.

Appendix A: Forward algorithm

In Appendices A–D we present details of our algorithms including forward and backward algorithms, which are used

for computing the normalization factor Z_X , gradient estimating algorithms, and Viterbi algorithm for inference.

The forward algorithm is used to compute the normalization factor Z_X efficiently by using the dynamic programming method. First, we denote

$$\alpha(y, t) = \sum_{S^t \in \Gamma_t^y} \text{Pol}(S^t) = \sum_{S^t \in \Gamma_t^y} \prod_{i=1}^q \Psi(s_{i-1}, s_i, X), \quad (30)$$

where $\Gamma_t^y = \{S = s_1, s_2, \dots, s_q\}$ is a set of all semi-Markov sequences, which have an original label sequence (y_1, y_2, \dots, y_t) with the last expected label is y . Thus, every $S^t = s_1, s_2, \dots, s_q \in \Gamma_t^y$ satisfies $s_q.e \leq t$ and $s_q.y = y$.

$$\gamma(y, t) = \sum_{S^t \in \Lambda_t^y} \text{Pol}(S^t) = \sum_{S^t \in \Lambda_t^y} \prod_{i=1}^q \Psi(s_{i-1}, s_i, X), \quad (31)$$

where $\Lambda_t^y = \{S = s_1, s_2, \dots, s_q\}$, is a set of all semi-Markov sequences, which have an original label sequence $(y_1, y_2, \dots, y_{t-1}, y_t = y)$. Therefore, every $S^t = s_1, s_2, \dots, s_q \in \Lambda_t^y$ satisfies $s_q.e = t$ and $s_q.y = y$. Let ϕ^t represent a special sequence of length t , which contains only “IA” labels. From (14) we have

$$\begin{aligned} Z_X &= \sum_{S^T} \text{Pol}(S^T) \\ &= \sum_y \sum_{S^T \in \Gamma_T^y} \text{Pol}(S^T) + \text{Pol}(\phi^T) \\ &= \sum_y \alpha(y, T) + e^{G_{IA}(IA, 1, T)}. \end{aligned} \quad (32)$$

To compute $\alpha(y, t)$ efficiently, we note that

$$\begin{aligned} \alpha(y, t) &= \sum_{S^t \in \Gamma_t^y} \text{Pol}(S^t) \\ &= \sum_{S^{t-1} \in \Gamma_{t-1}^y} \text{Pol}(S^{t-1} \oplus IA) + \sum_{S^t \in \Lambda_t^y} \text{Pol}(S^t), \end{aligned} \quad (33)$$

where $S^{t-1} \oplus IA$ denotes the concatenation of a label “IA” to the end of the original sequence of S^{t-1} . It is easy to see that

$$\text{Pol}(S^{t-1} \oplus IA) = \text{Pol}(S^{t-1})e^{G_{IA}(IA, t, t)}. \quad (34)$$

From (33) and (34) we have

$$\alpha(y, t) = \alpha(y, t - 1)e^{G_{IA}(IA, t, t)} + \gamma(y, t). \tag{35}$$

Derive $\gamma(y, t)$ from (31) we see that

$$\begin{aligned} \gamma(y, t) &= \sum_{S^t \in A_t^y} \text{Pol}(S^t) \\ &= \sum_{d=1}^D \sum_{S^{t-d}} \text{Pol}(S^{t-d} \oplus (y, t - d + 1, t)), \end{aligned} \tag{36}$$

where $S^{t-d} \oplus (y, t - d + 1, t)$ represents the appending of d labels y to the original sequence of S^{t-d} . In case S^{t-d} contains at least one state, we can assume that (y^*, b, e) is the last state of S^{t-d} , then we have

$$\begin{aligned} \text{Pol}(S^{t-d} \oplus (y, t - d + 1, t)) \\ = \text{Pol}(S^{t-d})e^{w^{Tr}(y^*, y) + G(y, t-d+1, t)}. \end{aligned} \tag{37}$$

In the other case, $S^{t-d} = \emptyset$ or its original sequence comprises of only "IA". It is clear that

$$\begin{aligned} \text{Pol}(S^{t-d} \oplus (y, t - d + 1, t)) \\ = e^{G_{IA}(IA, 1, t-d)} e^{G(y, t-d+1, t)}. \end{aligned} \tag{38}$$

From (36), (37), (38) we conclude that

$$\begin{aligned} \gamma(y, t) &= \sum_{d=1}^D \left(\sum_{y'} \alpha(y', t - d) e^{w^{Tr}(y', y) + G(y, t-d+1, t)} \right. \\ &\quad \left. + e^{G_{IA}(IA, 1, t-d) + G(y, t-d+1, t)} \right). \end{aligned} \tag{39}$$

Obviously in (39) only $\alpha(y', t - d)$ and $w^{Tr}(y', y)$ depend on y' therefore by pre-calculating

$$\lambda(y, t) = \sum_{y'} \alpha(y', t) e^{w^{Tr}(y', y)}, \tag{40}$$

we have

$$\begin{aligned} \gamma(y, t) &= \sum_{d=1}^D (\lambda(y, t - d) e^{G(y, t-d+1, t)} \\ &\quad + e^{G_{IA}(IA, 1, t-d) + G(y, t-d+1, t)}). \end{aligned} \tag{41}$$

Based on (32), (35), (40), and (41) the forward algorithm can be implemented as in below pseudo-code.

Algorithm 1: Forward algorithm for calculating Z_X

Forward

```

for  $t = 1$  To  $T$  do
  for  $y = 1$  To  $StateNum$  do
     $\alpha[y][t] = 0$ 
     $\gamma[y][t] = 0$ 
     $\lambda[y][t] = 0$ 
    for  $d = 1$  To  $D$  do
      if  $t - d + 1 > 0$  then
         $\gamma[y][t] +=$ 
         $\lambda[y][t - d] e^{G(y, t-d+1, t)}$ 
         $\gamma[y][t] +=$ 
         $e^{G_{IA}(IA, 1, t-d) + G(y, t-d+1, t)}$ 
      else
         $\perp$  Break
      if  $t > 1$  then
         $\alpha[y][t] =$ 
         $\alpha[y][t - 1] e^{G_{IA}(IA, t, t)} + \gamma[y][t]$ 
      else
         $\perp$   $\alpha[y][t] = \gamma[y][t]$ 
      for  $y' = 1$  To  $StateNum$  do
         $\lambda[y][t] = \lambda[y][t] + \alpha[y'][t] e^{w^{Tr}(y', y)}$ 
     $Z_X = e^{G_{IA}(IA, 1, T)}$ 
  for  $y = 1$  To  $StateNum$  do
     $\perp$   $Z_X = Z_X + \alpha(y, T)$ 

```

end

Appendix B: Backward algorithm

Similarly to the forward algorithm, let we denote

$$\begin{aligned} \beta(y, t) &= \sum_{S^{T-t+1} \in \Omega_t^y} \text{Pol}(S^{T-t+1}) \\ &= \sum_{S^{T-t+1} \in \Omega_t^y} \prod_{i=1}^q \Psi(s_{i-1}, s_i, X), \end{aligned} \tag{42}$$

where $\Omega_t^y = \{S = s_1, s_2, \dots, s_q\}$ is a set of all semi-Markov sequences, which have an original label sequence $(y_t, y_{t+1}, \dots, y_T)$ with the first expected label is y .

$$\eta(y, t) = \sum_{S^{T-t+1} \in \mathcal{Y}_t^y} \text{Pol}(S^{T-t+1}), \tag{43}$$

where $\mathcal{Y}_t^y = \{S = s_1, s_2, \dots, s_q\}$ is a set of all semi-Markov sequences, which have an original label sequence $(y_t = y, y_{t+1}, \dots, y_T)$. Follow similar steps in forward algorithm we come up with

$$\beta(y, t) = \beta(y, t + 1) e^{G_{IA}(IA, t, t)} + \eta(y, t), \tag{44}$$

Algorithm 2: Backward algorithm for calculating Z_X **Backward**

```

for  $t = T$  Down To 1 do
  for  $y = 1$  To  $StateNum$  do
     $\beta[y][t] = 0$ 
     $\eta[y][t] = 0$ 
     $\zeta[y][t] = 0$ 
    for  $d = 1$  To  $D$  do
      if  $t + d - 1 \leq T$  then
         $\eta[y][t] +=$ 
         $\zeta[y][t + d]e^{G(y,t,t+d-1)}$ 
         $\eta[y][t] +=$ 
         $e^{G(y,t,t+d-1)+G_{IA}(IA,t+d,T)}$ 
      else
         $\perp$  Break
      if  $t < T$  then
         $\beta[y][t] =$ 
         $\beta[y][t + 1]e^{G_{IA}(IA,t,t)} + \eta[y][t]$ 
      else
         $\perp$   $\beta[y][t] = \eta[y][t]$ 
      for  $y' = 1$  To  $StateNum$  do
         $\zeta[y][t] = \zeta[y][t] + \beta[y'][t]e^{w^{Tr}(y,y')}$ 
     $Z_X = e^{G_{IA}(IA,1,T)}$ 
  for  $y = 1$  To  $StateNum$  do
     $\perp$   $Z_X = Z_X + \beta(y, 1)$ 

```

end

$$\begin{aligned}
\eta(y, t) &= \sum_{d=1}^D \left(\sum_{y'} \beta(y', t+d) e^{w^{Tr}(y,y') + G(y,t,t+d-1)} \right. \\
&\quad \left. + e^{G(y,t,t+d-1) + G_{IA}(IA,t+d,T)} \right) \\
&= \sum_{d=1}^D \left(\zeta(y, t+d) e^{G(y,t,t+d-1)} \right. \\
&\quad \left. + e^{G(y,t,t+d-1) + G_{IA}(IA,t+d,T)} \right), \quad (45)
\end{aligned}$$

where

$$\zeta(y, t) = \sum_{y'} \beta(y', t) e^{w^{Tr}(y,y')}. \quad (46)$$

The following pseudo-code illustrates how the backward algorithm can be implemented.

Appendix C: Gradient estimation

The goal of parameter estimation is to choose appropriate values for the model weights (w^{Tr} , w^D , and w^O) so that the likelihood of the observation data $P(S|X)$ is maximized.

Take the logarithm form of $P(S|X)$ we have

$$\begin{aligned}
L(S|X) &= \sum_{i=1}^P (Q^{Tr}(s_{i-1}, s_i, X) + Q^D(s_{i-1}, s_i, X) \\
&\quad + Q^O(s_{i-1}, s_i, X)) - \log(Z_X). \quad (47)
\end{aligned}$$

To find the optimal parameter values w^* we have to solve $\frac{dL}{dw^*} = 0$. From (47) we know that

$$\frac{dL}{dw^*} = \sum_{i=1}^P \frac{dQ^*(s_{i-1}, s_i, X)}{dw^*} - \frac{1}{Z_X} \frac{dZ_X}{dw^*}. \quad (48)$$

Herein we use Q^* and w^* to refer to any kind of the potential function and weight (* can be D , Tr , or O). Computing the first term of the right side in (48) is trivial, Z_X is calculated by using forward or backward variables. Therefore, here we mainly focus on evaluating $\frac{dZ_X}{dw^*}$ for different kind of weights. From (14) and (23) we have

$$\frac{dZ_X}{dw^*} = \sum_{S^T} \left(\left(\sum_{i=1}^P \frac{dQ^*(s_{i-1}, s_i, X)}{dw^*} \right) \prod_{i=1}^P \Psi(s_{i-1}, s_i, X) \right). \quad (49)$$

C.1 Gradient of the transition weight

Since

$$\frac{dQ^{Tr}(s_{i-1}, s_i, X)}{dw^{Tr}(y', y)} = \delta(s_{i-1}.y = y', s_i.y = y), \quad (50)$$

it brings about that

$$\frac{dZ_X}{dw^{Tr}(y', y)} = \sum_{t=1}^T \sum_{S^{Tr} \in \Lambda_t^{y'} \oplus \Omega_{t+1}^y} \prod_{i=1}^P \Psi(s_{i-1}, s_i, X), \quad (51)$$

where each $S^T = s_1, s_2, \dots, s_P \in \Lambda_t^{y'} \oplus \Omega_{t+1}^y$ can be defined as the concatenation of two sub sequences $S_{prev}^t \in \Lambda_t^{y'}$ and $S_{post}^{T-t} \in \Omega_{t+1}^y$. Therefore

$$\frac{dZ_X}{dw^{Tr}(y', y)} = \sum_{t=1}^T \gamma(y', t) \beta(y, t+1) e^{w^{Tr}(y', y)}. \quad (52)$$

C.2 Gradient of the duration weight

From the definition of the duration potential function, it is obvious that

$$\begin{aligned}
&\frac{dQ^D(s_i, s_{i-1}, X)}{dw^D(y)} \\
&= \sum_{y,d} \delta(s_i.y = y, s_i.e - s_i.b + 1 = d) \frac{(d - m_y)^2}{2\sigma_y^2}. \quad (53)
\end{aligned}$$

As a result

$$\frac{dZ_X}{dw^D(y)} = \sum_{d=1}^D \sum_{t=1}^T \frac{(d - m_y)^2}{2\sigma_y^2} \sum_{S^T \in \chi_y^{d,t}} \prod_{i=1}^P \psi(s_{i-1}, s_i, X), \tag{54}$$

where $\chi_y^{d,t} = \{S = s_1, \dots, s_P\}$ is a set of all semi-Markov sequences whose original sequences contain d continuous labels y from time t . Let

$$\theta(y, t, d) = \sum_{S^T \in \chi_y^{d,t}} \prod_{i=1}^P \psi(s_{i-1}, s_i, X). \tag{55}$$

Obviously, each $S^T \in \chi_y^{d,t}$ can be represented as a concatenation

$$S^T = S_{prev}^{t-1} \oplus (y, t, t + d - 1) \oplus S_{post}^{T-t-d+1}, \tag{56}$$

where

$$S_{prev}^{t-1} \in \bigcup_{y'} \Gamma_{t-1}^{y'} \cup \{\phi^{t-1}\}, \tag{57}$$

and

$$S_{post}^{T-t-d+1} \in \bigcup_{y^*} \Omega_{t+d}^{y^*} \cup \{\phi^{T-t-d+1}\}. \tag{58}$$

Equations (55), (56), (57), and (58) imply that

$$\begin{aligned} \theta(y, t, d) &= (\lambda(y, t - 1)\zeta(y, t + d)e^{G(y,t,t+d-1)} \\ &\quad + \zeta(y, t + d)e^{G_{IA}(IA,1,t-1)+G(y,t,t+d-1)} \\ &\quad + \lambda(y, t - 1)e^{G(y,t,t+d-1)+G_{IA}(IA,t+d,T)} \\ &\quad + e^{G_{IA}(IA,1,t-1)+G(y,t,t+d-1)+G_{IA}(IA,t+d,T)}). \end{aligned} \tag{59}$$

Using $\theta(y, t, d)$ the gradient is calculated as

$$\frac{dZ_X}{dw^D(y)} = \sum_{d=1}^D \sum_{t=1}^T \frac{(d - m_y)^2}{2\sigma_y^2} \theta(y, t, d). \tag{60}$$

C.3 Gradient of the observation weight

To estimate the gradient of the observation weight, we consider two cases. In the first case, we handle the observation weights of expected labels. From (27), (28), and (29) we can assert that

$$\frac{dQ^O(s_{i-1}, s_i, X)}{dw^O(y, o)} = \sum_{k=s_i.b}^{s_i.e} \delta(s_i \cdot y = y, x_k = o). \tag{61}$$

Combining (61) and the definition of θ in (55) leads to

$$\frac{dZ_X}{dw^O(y, o)} = \sum_{\substack{k,t,d \\ k \in [t,t+d-1]}} \theta(y, t, d)\delta(x_k = o). \tag{62}$$

Similarly, we come up with estimation equations in case of irrelevant labels

$$\frac{dZ_X}{dw^O(IA, o)} = \sum_{t=1}^T v(t)\delta(x_t = o), \tag{63}$$

where

$$v(t) = \sum_{S^T} \prod_{i=1}^P \psi(s_{i-1}, s_i, x), \tag{64}$$

sequence $S^T = s_1, s_2, \dots, s_P$ does not contain any expected activities at time t . So, S^T can be decomposed as

$$S^T = S_{prev}^{t-1} \oplus IA \oplus S_{post}^{T-t}, \tag{65}$$

where

$$S_{prev}^{t-1} \in \bigcup_{y'} \Gamma_{t-1}^{y'} \cup \{\phi^{t-1}\}, \tag{66}$$

and

$$S_{post}^{T-t} \in \bigcup_y \Omega_{t+1}^y \cup \{\phi^{T-t}\}. \tag{67}$$

Finally we have

$$\begin{aligned} v(t) &= \left(\sum_{y'} \sum_y \alpha(y', t - 1)\beta(y, t + 1)e^{w^{Tr}(y',y)+G_{IA}(IA,t,t)} \right. \\ &\quad + \alpha(y', t - 1)e^{G_{IA}(IA,t,T)} + \beta(y, t + 1)e^{G_{IA}(IA,1,t)} \\ &\quad \left. + e^{G_{IA}(IA,1,T)} \right). \end{aligned} \tag{68}$$

After computing the derivatives, many convex optimization techniques can be applied [3]. In our solution, we applied a simple stochastic gradient ascent method to finding the optimal result. To avoid over-fitting, we use L1 regularizer which is $-\frac{1}{2}W^T W$. Therefore, the actual target function is

$$\begin{aligned} L(S|X) &= \sum_{i=1}^P (Q^{Tr}(s_{i-1}, s_i, X) + Q^D(s_{i-1}, s_i, X) \\ &\quad + Q^O(s_{i-1}, s_i, X)) \\ &\quad - \log(Z_X) - \varepsilon \frac{1}{2}W^T W, \end{aligned} \tag{69}$$

where ε is a smoothing constant, which is manually estimated.

Appendix D: Inference using Viterbi algorithm

Inference in our semi-CRF is done by using Viterbi algorithm with a complexity of $O(TM^2D)$. Our target is to find the best matched sequence Y given an input X so that $P(Y|X)$ is maximized. Let we denote

$$\delta(y, t) = \max \log (P(S^t = s_1, s_2, \dots, s_q | x_1, x_2, \dots, x_t)), \quad (70)$$

$S^t = s_1, s_2, \dots, s_q$ has the last expected activity is y , or equivalently $s_{q \cdot e} \leq t$ and $s_{q \cdot y} = y$.

$$\delta(y, t) = \max_d \begin{cases} A = \delta(y, t-1) + G_{IA}(IA, t, t), \\ B = G_{IA}(IA, 1, t-d) \\ \quad + G(y, t-d+1, t), \\ \delta(y', t-d) + w^{Tr}(y', y) \\ \quad + G(y, t-d+1, t). \end{cases} \quad (71)$$

For backtracking we use $\Delta^{State}(y, t)$ and $\Delta^{Duration}(y, t)$ to store the previous trace of $\delta(y, t)$ as following

$$\Delta^{State}(y, t) = \begin{cases} y & \text{if } \delta(y, t) = A, \\ IA & \text{if } \delta(y, t) = B, \\ y' & \text{otherwise,} \end{cases} \quad (72)$$

Algorithm 3: Viterbi algorithm for tracking the best matched sequence

Step 1. Initialization

```

 $y^* = \operatorname{argmax} \delta(y, T)$ 
 $t = T$ 
 $y = y^*$ 
 $i = 1$ 

```

end

Step 2. Backtracking

```

while  $y \neq IA$  and  $t > 0$  do
  while  $\Delta^{Duration}(y, t) = 0$  do
     $t = t - 1$ 
     $s_i \cdot y = y$ 
     $s_i \cdot e = t$ 
     $s_i \cdot b = t - \Delta^{Duration}(y, t) + 1$ 
     $i = i + 1$ 
     $t = t - \Delta^{Duration}(y, t)$ 
   $y = \Delta^{State}(y, t)$ 
 $P = i - 1$ 

```

end

Step 3. Finalization

```

 $y_1, y_2, \dots, y_T =$ 
 $OriginalSequence(s_P, s_{P-1}, \dots, s_1)$ 

```

end

$$\Delta^{Duration}(y, t) = \begin{cases} 0 & \text{if } \delta(y, t) = A, \\ d & \text{if } \delta(y, t) = B, \\ d & \text{otherwise.} \end{cases} \quad (73)$$

Using $\delta(y, t)$, $\Delta^{State}(y, t)$ and $\Delta^{Duration}(y, t)$ we can follow the below steps to track the optimized path.

References

- Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: Proceedings of the 2nd international conference on pervasive computing, vol 3001, pp 1–17
- Blanke U, Schiele B (2009) Daily routine recognition through activity spotting. In: Proceedings of the 4th international symposium on location and context-awareness, vol 5561, pp 192–206
- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
- Brdiczka O, Maisonnasse J, Reignier P, Crowley JL (2009) Detecting small group activities from multimodal observations. Appl Intell 30:47–57
- Doherty MK (2007) Gene prediction with conditional random fields. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
- Favela J, Tentori M, Castro LA, Gonzalez VM, Moran EB, Martínez-García AI (2007) Activity recognition for context-aware hospital applications: issues and opportunities for the deployment of pervasive networks. Mob Netw Appl 12:155–171
- Huynh T, Blanke U, Schiele B (2007) Scalable recognition of daily activities with wearable sensors. In: Proceedings of the 3rd international symposium on location and context-awareness, vol 4718, pp 50–67
- Huynh T, Fritz M, Schiele B (2008) Discovery of activity patterns using topic models. In: Proceedings of the 10th international conference on ubiquitous computing, vol 344, pp 10–19
- Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning, pp 282–289
- Lampe M, Strassner M, Fleisch E (2004) A ubiquitous computing environment for aircraft maintenance. In: Proceedings of ACM symposium on applied computing, pp 1586–1592
- Liao L, Choudhury T, Fox D, Kautz HA (2007) Training conditional random fields using virtual evidence boosting. In: Proceedings of the 20th international joint conference on artificial intelligence, pp 2530–2535
- Liao L, Fox D, Kautz HA (2007) Extracting places and activities from gps traces using hierarchical conditional random fields. Int J Rob Res 26:119–134
- Lukowicz P, Starner TE, Troster G, Ward JA (2006) Activity recognition of assembly tasks using body-worn microphones and accelerometers. IEEE Trans Pattern Anal Mach Intell 28:1553–1567
- Najafi B, Aminian K, Loew F, Blanc Y, Robert PA (2002) Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly. IEEE Trans Biomed Eng 49:843–851
- Okanohara D, Miyao Y, Tsuruoka Y, Tsujii J (2006) Improving the scalability of semi-Markov conditional random fields for named entity recognition. In: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics, pp 465–472
- Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, vol 77, pp 257–286

17. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. In: Proceedings of the 20th national conference on artificial intelligence, vol 20, pp 1541–1546
18. Sánchez D, Tentori M, Favela J (2008) Activity recognition for the smart hospital. *IEEE Intell Syst* 23:50–57
19. Sarawagi S, Cohen W (2005) Semi-Markov conditional random fields for information extraction. In: Proceedings of the 2004 conference on neural information processing systems, pp 1185–1192
20. Suutala J, Pirttikangas S, Röning J (2007) Discriminative temporal smoothing for activity recognition from wearable sensors. In: Proceedings of the 4th international symposium on ubiquitous computing systems, vol 4836, pp 182–195
21. Tapia EM, Intille SS, Larson K (2004) Activity recognition in the home setting using simple and ubiquitous sensors. In: Proceedings of the 2nd international conference on pervasive computing, vol 3001, pp 158–175
22. Truyen TT, Phung DQ, Bui HH, Venkatesh S (2009) Hierarchical semi-Markov conditional random fields for recursive sequential data. In: Proceedings of the 22nd annual conference on neural information processing systems, pp 1657–1664
23. Vail DL, Veloso MM, Lafferty JD (2007) Conditional random fields for activity recognition. In: Proceedings of the 6th international joint conference on autonomous agents and multi-agent systems, p 235
24. World Health Organization (2004) Global strategy on diet, physical activity and health. Technical report



La The Vinh received his B.S. and M.S. from Ha Noi University of Technology, Vietnam, in 2004 and 2007 respectively. Since September 2008, he has been working on his PhD degree at the Department of Computer Engineering at Kyung Hee University, Korea. His research interests include digital signal processing, pattern recognition and artificial intelligence. Email: vinhlt@oslab.khu.ac.kr



Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and PhD degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been affiliated with a director of Neo Medical ubiquitous-Life Care Information Technology Research Center, Kyung Hee University since 2006. He is a member of ACM and IEEE. Email: sylee@oslab.khu.ac.kr



Hung Xuan Le received BS degree in Computer Science from Hanoi University, Vietnam, in 2003, M.S., Ph.D degree in Computer Engineering from Kyung Hee University, Korea, in 2005 and 2008, respectively. From 2002 to 2003, he served as a software engineering at FPT corp., Vietnam. From Dec 2008 to Aug 2009, he was a research professor at Dept. of Computer Engineering, Kyung Hee University, Korea. Since Sept 2009, he has been a Postdoctoral Research Associate at Dept. of Electrical Engineering, University of South Florida, USA. His research interests are wireless sensor networks, cryptography and information security. Email: xhle@eng.usf.edu



Hung Quoc Ngo received his B.E. in Telecommunications from Ho-Chi-Minh City University of Technology, Vietnam, in 2003, with highest honors. He got his M.S. in Computer Engineering from Kyung Hee University, Korea, in 2005, with highest honors. His research interests include factor graphs and its applications in distributed optimization and inference in wireless sensor networks and machine learning. Email: nqhung@oslab.khu.ac.kr



Hyoung Il Kim received his MS in computer engineering from the Kyung Hee University, Korea in 1996. Currently, he is a Bada (Samsung Smartphone) platform engineer in the Mobile Communication Division at Samsung Electronics. His research interests are activity recognition for ubiquitous & mobile computing. Email: meeso.kim@samsung.com



Manhyung Han received his Bachelor and Master degrees in Dept. of Computer Engineering from Kyung Hee University, Korea in 2005 and 2007, respectively. Currently, he is working as a Ph.D degree in Dept. of Computer Engineering at Kyung Hee University. His research areas include ubiquitous computing, context-aware middleware, activity recognition and WSN. Email: smiley@oslab.khu.ac.kr



Young-Koo Lee got his B.S., M.S. and PhD in Computer Science from Korea advanced Institute of Science and Technology, Korea. He is a professor in the Department of Computer Engineering at Kyung Hee University, Korea. His research interests include ubiquitous data management, data mining, and databases. He is a member of the IEEE, the IEEE Computer Society, and the ACM. Email: yklee@khu.ac.kr