2A1-IOS-3b-1

# Activity Recognition on Multi-Sensor Data Streams Using Distinguishing Sequential Patterns

Rong-Sing Huang Been-Chian Chien

Department of Computer Science and Information Engineering
National University of Tainan
Tainan, Taiwan, R. O. C.

Activity recognition is an important task in the researches of ambient intelligence and smart environment. In real smart environments, detected signals of monitoring living space come from heterogeneous sensors generally. The intelligent awareness system can automatically provides the proper services to satisfy the requirements of users based on the recognized activity of users sensed by multi-sensor. The high accurate activity recognition is the basis of supporting high-quality service. However, the streaming data generated by multi-sensor are real time, continuous and variable. It is difficult for systems to archive high precise activity recognition from data streams of multiple sensors. The previous models on recognizing sequential data includes Hidden Markov model and Conditional Random Fields. This paper proposed a new activity recognition model for processing sequential data stream based on mining distinguishing sequential patterns. The general sequential patterns are on-line generated and counted first from the data streams and the minimal distinguishing sequential patterns are mined. Then, efficient and effective probabilistic recognizing methods and algorithms are developed for activity recognition. Two datasets, WSU and Kasteren, were used to test the proposed methods. The experimental results show that the proposed models have effective recognition rates in both of the activity level and the time slides level. The proposed model also provides a strong on-line recognition paradigm on multi-sensor data stream.

## 1. Introduction

An integration of technologies in pervasive computing and machine learning initiates the development of smart environment, such as aware home, smart space, and ambient intelligence. A smart environment is generally equipped with variant types of sensors in the space to detect the environmental status and recognize users' activities for supporting proper services to users. The problem of recognizing activities accurately from multiple sensors is one of important tasks in the applications of smart environment. As the success of a smart environment relies on the satisfaction of users' requirements, recognizing environmental status to fit users' intention is the critical technique for an intelligent awareness system. High accurate activity recognition schemes can increase serving quality and reliability of the system. A high-quality service system is the main objective of a smart environment.

For detecting events in a smart environment, different sensors are usually installed around the target space. The sensing data from multiple sensors generate a collection of multi-dimensional data streams. Activity recognition in a smart environment can be treated as the sequence classification problem on multi-sensor data streams within a specific period of time. There are several challenges for activity recognition in multi-sensor data streams. The first obstacle is the noise of operating sensors in the environment. The signal interference often occurs among sensors due to abnormal awareness or user's unintentional behavior. The second difficulty is that it is hard to separate the sensor data stream into activity fragments. Because the sensing data are received on-line and real-time, it is almost impossible to separate the activity precisely. The third problem is that the data sequences have lots of variation between long activities and short activities. Since the streaming data generated by multiple sensors are real-time, on-line and noisy, it is difficult for researchers to find a general model to solve such a problem.

Many activity recognition methods were proposed in the last decade. Most activity recognition schemes are based on Hidden Markov Model (HMM) and Conditional Random Fields (CRF), such as [Hsu 2010], [Kasteren 2008], [Singla 2010] and [Yakhnenko 2005]. Since the traditional sequential classification models using maximum likelihood to find the best parameters, the previous activity recognition methods performed well in off-line circumstance of activity labeling and training. However, the training time is getting growing while the streaming data is increasing. As the identification task are employed in on-line multi-sensor data streams, classification models may not be able to be retrained completely in time for catching up the last formulating model.

In this paper, we propose a new activity recognition model for processing sequential data stream based on mining distinguishing sequential patterns. The minimal distinguishing subsequence mining approach [Ji 2005] is modified and applied to find minimal distinguishing sequential patterns. After discovering the minimal distinguishing patterns, the probabilistic analysis and the sequence classification algorithms are developed to recognize activities from a sequence of multi-sensor data stream efficiently and effectively. The evaluation and experiments are tested on two well known data sets, WSU [WSU 2010] and Kasteren [Kasteren 2008]. The experimental results show that the proposed schemes can improve the effectiveness both of the time slice accuracy and class accuracy in raw data. Furthermore, it is also effective while the raw data are

Contact: Been-Chian Chien, Department of Computer Science and Information Engineering, National University of Tainan, 33, Sec. 2, Su-Lin Street, Tainan, 70005, Taiwan. Tel: +886-62606123 ext.7707. Email: bcchien@mail.nutn.edu.tw

preprocessed by the 'change point' sampling method or the 'last' filtering techniques. The results of comparing with HMM are drawn and the properties of the proposed sequence classification method are discussed.

The rest of this paper is organized as follows. In Section 2, the related researches on sequence classification techniques for activity recognition are discussed. The new method based on distinguishing pattern and the proposed activity classification algorithms are presented in Section 3. Section 4 describes the evaluation of the proposed new approaches and discusses the experimental results. Concluding remarks and further work are made in the final section.

## 2. Related Work

Sequence classification is one of the primitive techniques of recognizing activity in streaming data. Generally, the sequence classification methods can be divided into three categories: model based classification, feature based classification and distance based classification [Xing 2010]. We will give a brief review on sequence classification methods of the three categories in the following subsections.

### 2.1 Model based sequence classification

The model based sequence classification method assumes that sequences in a class are generated by an underlying generative probabilistic model. The most common models include Naïve Bayes classifier, Markov Model (MM), Hidden Markov Model (HMM), and Conditional Random Fields (CRF). The model is usually defined on an assumption of probability distribution described by a set of parameters. The objective of training phase is to learn the optimized parameters of the model. Then, the classification phase can assign a unknown sequence to the class with highest likelihood or probability.

Naïve Bayes is usually applied to the application which the sequences are independent of each other, e.g. text classification [Lewis 1998]. The MM and the HMM are generally used to model the classification tasks having dependence among elements in their sequences. For example, Yakhnenko et. al. apply a K-order Markov Model to classify protein and text sequences [Yakhnenko 2005]. Kasteren et. al. apply Hidden Markov Model to recognize activities in smart environment [Kasteren 2008].

### 2.2 Feature based sequence classification

In the feature based classification, a sequence datum is first transformed into a multi-dimensional feature vector. Then, conventional classification methods, like decision trees, neural network, SVM, etc., are applied to accomplish the training and classifying tasks. As a consequence of applying conventional classification methods, the extracting and selecting effective features from sequences become the most important task while classifying sequences.

Several feature extraction methods were proposed to find meaningful patterns from data sequences. Chuzhanova et. al. apply k-grams to generate all possible subsequence in training set, and apply Gamma test to select more informative feature set [Chuzhanova 1988]. Lest et. al. propose an Apriori based feature mining method to find distinguished feature patterns [Lesh 1999]. Each selected features must be short subsequences which satisfy the following rules: 1) It is frequent. 2) It should be distinctive at least one class. 3) It cannot contain redundant features. After features being selected, Winnow [Littlestone 1988] and Naïve Bayes classifier are applied to classify sequences. Ji et. al. proposed an effective algorithm to mine minimal distinguishing subsequences with maximal gap constraint [Ji 2005]. The found subsequence patterns are frequent in one class and infrequent in other classes.

### 2.3 Distance based sequence classification

The distance based classification methods must define a distance function to measures the similarity between two sequences. After obtaining the difference of similarity between a pair of sequences, some distance based classifiers, like $k$ nearest neighbor classifier (KNN) and SVM with local alignment kernel, are used to classify sequence data.

The measure of distance function is the critical part for the distance based sequence classification. For classifying time series, Euclidean distance and the dynamic time warping distance (DTW) [Keogh 2000] are adopted widely in various applications. For symbolic sequences, the alignment based distance measures are usually adopted [Kaján 2006]. Many variants based on the alignment method are also developed, such as global alignment, local alignment, and region alignment.

## 3. The Proposed Methods

The solution of activity recognition on data streams proposed in this section is a kind of feature based classification method. First, we will give the problem precisely in formal notation. Then, we depict the approach of mining the minimal distinguishing patterns which are frequent in the class but infrequent in other classes. Finally, the probabilistic based algorithms are designed to separate the data sequence and classify the activity.

### 3.1 Notation

The activity recognition in multi-sensor environment can be formalized as follows. Let $s_i$ be a multi-sensor data vector and $S = s_1s_2\ldots s_t$ is a multi-sensor streaming data. The notation is $s_i \in S$. The $s_t$ is the last coming data. For a finite class set of activities $C=\{c_1, c_2, \ldots, c_K\}$, $c_k$ is the class label of $k$th activity, $1 \le k \le K$. The labeled streaming data are represented as $(s_i, c_k)$, where $1 \le i \le t$ and $1 \le k \le K$. A subsequence containing $s_i$ having the same activity label in its neighbor will partition the streaming data into activity subsequence denoted as $(S_l, c_k)$. The $S_l$ be a subsequence labeled by activity $c_k$, and $|S_l|$ is the length of the subsequence $S_l$.

The goal of activity recognition on the data streams is to construct an effective recognizer to classify an on-line multi-sensor sequence based on the set of labeled training activity subsequence $(S_l, c_k)$, $1 \le l \le n$ and $1 \le k \le K$, where $n$ is the number of sequence data in training set. Thus, the problem of activity recognition on the data streams is not only classifying the sequence accurately but also partitioning the streaming sequence effectively.

### 3.2 Mining minimal distinguishing patterns

To find the features of recognizing activities, we describe the proposed mining approach based on [Ji 2005] to get minimal distinguishing patterns in this subsection.

The distinguishing patterns found in [Ji 2005] have two characteristics. The first is that the distinguishing patterns are subsequences allowing gaps inside the sequence. The second is the distinguishing patterns are in unlimited length. Since the length of distinguish patterns used in this task generally do not need too long, we simplify the algorithm to find the patterns we need.

First, the minimal distinguishing pattern is defined as follows: A sequence pattern *pat* is called distinguishing pattern if *pat* can identify unique activity among distinct activities. Further, while none of the subsequence in the pattern *pat* can identify the same activity, the *pat* is one of the minimal distinguishing patterns. Next, a long minimal distinguishing pattern may be too specific for classifiers to identify the correct activity. Therefore, the maximal prefix length (MPL) are defined and specified to restrict the size of a distinguish pattern.

Let $T$ be the training set of activity sequences $(S_l, c_k)$, $1 \le l \le n$ and $1 \le k \le K$. **S** is the set of all possible multi-sensor data. We also define two sets, *MDS* and *IS*, to be the set of minimal distinguishing patterns and the set of indistinguishable patterns, respectively. The minimal distinguishing pattern mining algorithm is depicted in Figure 1. The initial values of $MDS = \{\}$, $IS = \{\}$, and *pat* is empty string. The length of *pat* is increased one by one. If the current *pat* can not be one of the distinguishing patterns, the *pat* will grow one more multi-sensor data until the length of *pat* is larger than MPL.

```
Algorithm Find minimal distinguishing patterns
Input:   T: training set
         pat: sequence pattern
         S: the all possible multi-sensor data
Output:  MDS: the set of minimal distinguishing patterns
         IS: the set of indistinguishable patterns
Min_distinguish_pattern(pat)
{
  for s_i ∈ S
  {
     pat = concatenate(s_i, pat);
     if (pat is a distinguishing pattern in T)
        MDS = MDS ∪ pat;
     else
     {  IS = IS ∪ pat;
        if length(pat) < MPL
           Min_distinguish_pattern(pat);
     }
}
```

Figure 1. The minimal distinguishing patterns mining algorithm.

## 3.3 Sequence classification

After mining minimal distinguishing patterns, a probabilistic sequence classification scheme is developed by adopting the set *MDS* to classify the activity class for each on-line generated multi-sensor streaming data $s_i$. We introduce the proposed sequence classification method in the following.

Let $\boldsymbol{D}_k \in MDS$ be the sets of minimal distinguishing patterns being able to recognize the $k$th activity $c_k$ for $1 \le k \le K$. Assume that $S = s_1 s_2 \ldots s_i$ be the multi-sensor data stream and $s_i$ is the last signal we received. The procedure of the sequence classification contains three main stages.

The first stage uses the sets of minimal distinguishing patterns $\boldsymbol{D}_k \in MDS$ to identify the activity. In this stage, the sequence will be looked forward until the length of pattern exceeding MPL if the $s_i$ cannot decide the activity exactly. A delaying policy is also employed in this stage. While the forward sequence exceeds MPL and the activity is still not able to be recognizes by minimal distinguishing pattern. The current signal data $s_i$ will not be assigned the activity label instantly. The latent identification is allowed at this time if the number of undecided latent data is less than the specified length of delay.

Once any activity in sequence cannot be recognized exactly in the first stage, the probabilistic decision strategy would be further considered. For the unidentified stream data, the second stage of the procedure handles the idle case in which the values of $s_i$, $s_{i+1}$, … are the same and keep repeating. While the stream data keep the same data, it means that the user is idle. At this moment, the identified activity label at the last decision will be assigned to all the present undecided stream data.

In the third stage, we focus on using probabilistic estimation functions to recognize the activity and assign to the undecided stream data. Since the minimal distinguishing pattern in the first stage cannot identify the activity, this stage must take *IS*, the set of indistinguishable patterns, to solve this problem. Before introducing the probabilistic estimation functions, some notations are first defined as follows.

Let $p_j \in IS$ be the indistinguishable patterns obtained from the training set $T$ and $|IS| = m$. We assume that $x_{jk}$ is the number of patterns $p_j$ occurring in the training set $T$ with activity $c_k$. For a stream data $S$, if $s_t$ is the last unlabeled multi-sensor data, $x^l_{jk}$ is denoted as the number $x_{jk}$ whose $p_j$ matches the corresponding unlabelled stream data $s_{(t-l-1)} \ldots s_{(t-1)}\, s_t$ of length $l$.

Then, two different probabilistic estimation functions are introduced to compute the probability of each activity class, as follows:

$$fp_1(s_t \mid c_k) = \frac{\sum_{l=1}^{MPL} [l \times (x^l_{jk} \Big/ \sum_{k=1}^{K} x^l_{jk})]}{\sum_{l=1}^{MPL} l}, \text{ for } 1 \le k \le K.$$

$$fp_2(s_t \mid c_k) = \frac{\sum_{l=1}^{MPL} [l \times (x^l_{jk} \Big/ \sum_{k=1}^{K} x^l_{jk}) \times (x^l_{jk} \Big/ \sum_{(s_l, c_k) \in T} |S_l|)]}{\sum_{l=1}^{MPL} l}, \ 1 \le k \le K$$

The estimation function $fp_1$ refers to the percentage of pattern $p_i$ in the activity class and weighting as the length of individual pattern. In addition to the above conditions, the estimation function $fp_2$ considers the ratio of pattern $p_i$ appearing in the stream data with activity $c_k$ of the training set. After computing the estimation function of each activity, we assign the label of activity with maximum probability to all unlabeled stream data before $s_i$. The detailed sequence classification algorithm is shown in Figure 2.

```
Algorithm Sequence Classification
Input: s[], MPL, Delay
Output: The activity label of each s[].
{ sep = 0;
  pos = 1;
  un_pos = pos;
  d = Delay-1;
  for ( ; ; )
    { C_k = Check_DisPattern(s[], sep, pos);
      while ( C_k == 0 and d > 0)
      {  pos = pos + 1;
         C_k = Check_DisPattern(s[], sep, pos);
         d = d -1;
      }
      if (C_k ≠ 0)
      { Label(C_k, un_pos, pos);
        d = Delay - 1;
        if (C_k ≠ Activity of s[un_pos-1])      // separate activity
           sep = un_pos – 1;
        un_pos = pos + 1;
      }
      else   //  C_k == 0
      {  C_k = Find_RepPattern(s[], un_pos);
         if (C_k == 0)
            C_k = argmax_k{ fp(s[],c_k)} ;
         Label(C_k, un_pos, un_pos);
         if (C_k ≠ Activity of s[un_pos-1])      // separate activity
            sep = un_pos – 1;
         un_pos = un_pos + 1;
      }
      pos = pos + 1;
    }
}

Check_DisPattern(s[], sep, pos)
{
   i = pos – sep – MPL + 1;
   if  ( i < 0)
      i = 0;
   for ( i; sep + i <= pos; i++)
       if  ( s[sep + i, pos] in D_k )
          return Activity k;
      else
          return 0;
}

Find_RepPattern(s[], sep, i)
{
   if  ( (i –sep)≧MPL) and s[i – MPL+1, i] is unchanged)
      return Activity k of s[i-1];
   else
      return 0;
}
```

Figure 2. The sequence classification algorithm.

## 4. Experimental results

The evaluation of the proposed methods is described in this section. The purpose of the experiments is to evaluate the effectiveness of the proposed methods and make a comparison with the results of Hidden Markov Model (HMM) in [Kasteren 2008]. The testing task of activity recognition was done on two datasets, WSU dataset [WSU 2010] and Kasteren dataset (KD) [Kasteren 2008].

A daily life dataset generally contains various activities. However, because the lengths of different activities have large disparity, the evaluation of the effectiveness should concern about the ratios both of the correcting prediction in time slice and activity. The experiments are evaluated by two measures: time slice accuracy and class accuracy. The time slice accuracy stands for the percentage of correctly classified streaming data of daily time slice. The class accuracy represents the average percentage of correctly classified time slices of each activity. The two measures are defined as follows:

$$timeslice\_accuracy = \frac{\sum_{i=1}^{N}(predict(i) = true(i))}{N},$$

$$class\_accuracy = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i=1}^{N_k}(predict(i) = true(i))}{N_k},$$

where ($predict(i)$ = $true(i)$) is a binary indicator given 1 when the predicting activity for streaming data $s_i$ at time $i$ is the same to the ground truth; otherwise, the value 0 is given. $N$ is the total number of time slices. $K$ is the number of activities. $N_k$ is the total number of time slices with activity $k$.

- **WSU dataset**

The WSU dataset recorded the sensor data streams that 24 volunteers are assigned to perform five activities in the smart environment where 41 sensors were installed. The data are collected for 13 days with the multiple digital sensors. The activities include making a phone call, washing hands, cooking, eating, cleaning, and others. The detailed information is shown in Table 1.

The experimental results of the WSU dataset are shown in Table 2, Table 3, Table 4, and Table 5. The results in the four tables include time slice accuracy and class accuracy of the estimation functions $fp_1$ and $fp_2$, respectively. The experimental results show that the two methods achieve high time slice accuracy but lower class accuracy.

Table 1. The activities in WSU dataset.

| Activities | # of instances | % of time |
|---|---|---|
| Others | | 97.47% |
| Make a phone call | 24 | 0.36% |
| Wash hands | 24 | 0.15% |
| Cook | 24 | 0.94% |
| Eat | 24 | 0.34% |
| Clean | 24 | 0.74% |

Table 2. The time slice accuracy of WSU with $fp_1$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.9912 | 0.9928 | 0.9927 | 0.9915 |
| MPL = 2 | 0.9912 | 0.9927 | 0.9927 | 0.9915 |
| MPL = 3 | 0.9913 | 0.9928 | 0.9928 | 0.9917 |
| MPL = 4 | 0.9913 | 0.9936 | 0.9946 | **0.9947** |
| MPL = 5 | 0.9913 | 0.9936 | 0.9946 | **0.9947** |
| MPL = 6 | 0.9913 | 0.9936 | 0.9946 | **0.9947** |

Table 3. The class accuracy of WSU with $fp_1$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.6719 | 0.7273 | 0.7444 | 0.7151 |
| MPL = 2 | 0.6751 | 0.7201 | 0.7428 | 0.7218 |
| MPL = 3 | 0.6767 | 0.7223 | 0.7455 | 0.7259 |
| MPL = 4 | 0.6767 | 0.7224 | **0.7458** | 0.7265 |
| MPL = 5 | 0.6767 | 0.7224 | **0.7458** | 0.7265 |
| MPL = 6 | 0.6767 | 0.7224 | **0.7458** | 0.7265 |

Table 4. The time slice accuracy of WSU with $fp_2$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.9912 | 0.9928 | 0.9927 | 0.9915 |
| MPL = 2 | 0.9913 | 0.9928 | 0.9927 | 0.9916 |
| MPL = 3 | 0.9914 | 0.9928 | 0.9928 | 0.9918 |
| MPL = 4 | 0.9914 | 0.9936 | 0.9946 | **0.9947** |
| MPL = 5 | 0.9914 | 0.9936 | 0.9946 | **0.9947** |
| MPL = 6 | 0.9914 | 0.9936 | 0.9946 | **0.9947** |

Table 5. The class accuracy of WSU with $fp_2$

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.6721 | 0.7279 | 0.7450 | 0.7157 |
| MPL = 2 | 0.6753 | 0.7208 | 0.7435 | 0.7224 |
| MPL = 3 | 0.6769 | 0.7229 | 0.7462 | 0.7266 |
| MPL = 4 | 0.6769 | 0.7230 | **0.7465** | 0.7271 |
| MPL = 5 | 0.6769 | 0.7230 | **0.7465** | 0.7271 |
| MPL = 6 | 0.6769 | 0.7230 | **0.7465** | 0.7271 |

The two estimation functions gain almost the same time slice accuracy. The highest time slice accuracy appears at the case of MPL ≥ 4 and Delay = 4. In the part of class accuracy, we found that the results of the estimation function $fp_2$ are generally better than the function $fp_1$. The highest class accuracy appears at the case of MPL ≥ 4 and Delay = 4.

- **Kasteren dataset**

The Kasteren dataset contains the data streams that a 26-year-old man living alone in a three-room apartment where 14 state-change sensors were installed. The data are collected for 28 days with the multiple sensors. The activities include leaving, toileting, showering, sleeping, breakfast, dinner, and others.

The detailed information of activities is shown in Table 6, the sensor readings are set to get sampling per 60 seconds. The time slice duration is long enough to discriminative and short enough to provide high accuracy labeling results. For give a fair evaluation on such a dataset, the 28-days dataset are separated into training set and test set using n-fold cross validation, which is that one full day is used to test and the other remaining days are used to train.

Table 6. Kasteren dataset.

| **Activities** | **# of instances** | **% of time** |
|---|---|---|
| Others | | 11.50% |
| Leaving | 34 | 56.40% |
| Toileting | 114 | 1.00% |
| Showering | 23 | 0.70% |
| Sleeping | 24 | 29.00% |
| Breakfast | 20 | 0.30% |
| Dinner | 10 | 0.90% |
| Drink | 20 | 0.20% |

This data set is further preprocessed by the 'change point' sampling and the 'last' filtering. The raw sensor representation gives a 1 when triggering, and a 0 otherwise. However, the preprocess of 'change point' sampling gives a 1 only while the sensor changing being detected. The 'last' sensor filtering then keeps 1 when a sensor is triggered at the last and the other sensors will be turned into 0 at the same time.

The experimental results on the three different data sets (raw, change point, last) are shown in Table 7 to Table 18. The tables from Table 7 to Table 12 are the results of the estimation function $fp_1$. We found that time slice accuracy is better than class accuracy for all different datasets in the two estimation functions. For the measure of time slice accuracy, the raw data is better than the change point data in general but the best case (0.7876, in MPL=2 and Delay =2) of the change point data is higher than the raw data. For the measure of class accuracy, the change point data is higher than the raw data no doubt. Nevertheless, the last data has the highest time slice accuracy and class accuracy while using the estimation function $fp_1$.

The tables from Table 13 to Table 18 are the results of the estimation function $fp_2$. We found that for the measure of time slice accuracy, the raw data is higher than the change point data in all cases. However, for the measure of class accuracy, on the contrary, the change point data is higher than the raw data. As the estimation function $fp_1$, the last data still has the highest time slice accuracy and class accuracy in the estimation function $fp_2$.

Although the estimation functions $fp_1$ and $fp_2$ are almost in a tie for the raw data, generally speaking, the effectiveness of the estimation function $fp_1$ is better than $fp_2$ except for the measure of class accuracy in the last data. The class accuracy of the last data is the highest especially. The class accuracy has 12% - 20% higher than the other data sets in average. The reason is that the last changed sensors is usually the time of activity starting or acting in the environment of multiple sensors. This characteristics show that our proposed scheme can determine the short patterns with activity changing effectively.

Finally, we compare the proposed scheme with the results of HMM proposed in [Kasteren 2008]. The comparison is given in Figure 3 and Figure 4. The results of the proposed methods are selected from the case of MPL = 3 and Delay = 3. Figure 3 is the time slice accuracy and Figure 4 is the class accuracy. The comparison reflects the fact that the proposed scheme has great improvement in time slice accuracy for raw data. However, the change point data only senses the point of signal changing, it is disadvantageous for find the patterns for a specific activity.

Table 7. Time slice accuracy for raw data of KD with $fp_1$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.7405 | 0.7411 | 0.7411 | 0.7410 |
| MPL = 2 | 0.7456 | 0.7463 | 0.7463 | 0.7460 |
| MPL = 3 | 0.7454 | 0.7608 | **0.7610** | 0.7607 |
| MPL = 4 | 0.7456 | 0.7566 | 0.7552 | 0.7552 |
| MPL = 5 | 0.7456 | 0.7565 | 0.7562 | 0.7571 |
| MPL = 6 | 0.7460 | 0.7571 | 0.7572 | 0.7569 |

Table 8. Class accuracy for raw data of KD with $fp_1$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.4515 | 0.4648 | 0.4671 | 0.4685 |
| MPL = 2 | 0.5290 | 0.5399 | 0.5416 | **0.5424** |
| MPL = 3 | 0.4864 | 0.5234 | 0.5250 | 0.5254 |
| MPL = 4 | 0.4852 | 0.5132 | 0.5168 | 0.5188 |
| MPL = 5 | 0.4857 | 0.5077 | 0.5137 | 0.5162 |
| MPL = 6 | 0.4865 | 0.5082 | 0.5170 | 0.5194 |

Table 9. Time slice accuracy for change point data of KD with $fp_1$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.5824 | 0.5825 | 0.5826 | 0.5827 |
| MPL = 2 | 0.7874 | **0.7876** | 0.7871 | 0.7860 |
| MPL = 3 | 0.7269 | 0.7490 | 0.7494 | 0.7493 |
| MPL = 4 | 0.7228 | 0.7436 | 0.7451 | 0.7451 |
| MPL = 5 | 0.7231 | 0.7440 | 0.7458 | 0.7453 |
| MPL = 6 | 0.7232 | 0.7472 | 0.7475 | 0.7483 |

Table 10. Class accuracy for change point data of KD with $fp_1$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.5055 | 0.5077 | 0.5083 | 0.5088 |
| MPL = 2 | 0.6820 | 0.6827 | **0.6888** | 0.6872 |
| MPL = 3 | 0.6241 | 0.6510 | 0.6556 | 0.6537 |
| MPL = 4 | 0.6023 | 0.6192 | 0.6137 | 0.6135 |
| MPL = 5 | 0.6091 | 0.6263 | 0.6221 | 0.6207 |
| MPL = 6 | 0.6126 | 0.6300 | 0.6262 | 0.6267 |

Table 11. Time slice accuracy for last data of KD with $fp_1$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | **0.9516** | 0.9513 | 0.9505 | 0.9496 |
| MPL = 2 | 0.9075 | 0.9081 | 0.9070 | 0.9056 |
| MPL = 3 | 0.9411 | 0.9391 | 0.9374 | 0.9358 |
| MPL = 4 | 0.9424 | 0.9409 | 0.9380 | 0.9363 |
| MPL = 5 | 0.9454 | 0.9439 | 0.9404 | 0.9343 |
| MPL = 6 | 0.9454 | 0.9452 | 0.9417 | 0.9403 |

Table 12. Class accuracy for last data of KD with $fp_1$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.6696 | 0.6815 | 0.6820 | 0.6810 |
| MPL = 2 | 0.6968 | 0.7046 | **0.7024** | 0.7005 |
| MPL = 3 | 0.6909 | 0.6987 | 0.6958 | 0.7003 |
| MPL = 4 | 0.6926 | 0.6985 | 0.6989 | **0.7027** |
| MPL = 5 | 0.6816 | 0.6838 | 0.6849 | 0.6876 |
| MPL = 6 | 0.6817 | 0.6898 | 0.6907 | 0.6970 |

Table 13. Time slice accuracy for raw data of KD with $fp_2$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.7421 | 0.7426 | 0.7425 | 0.7422 |
| MPL = 2 | 0.7397 | 0.7405 | 0.7404 | 0.7399 |
| MPL = 3 | 0.7450 | 0.7599 | **0.7601** | 0.7598 |
| MPL = 4 | 0.7455 | 0.7555 | 0.7545 | 0.7546 |
| MPL = 5 | 0.7455 | 0.7555 | 0.7560 | 0.7569 |
| MPL = 6 | 0.7452 | 0.7562 | 0.7571 | 0.7573 |

Table 14. Class accuracy for raw data of KD with $fp_2$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.4821 | 0.4939 | 0.4948 | 0.4941 |
| MPL = 2 | 0.4922 | 0.5024 | 0.5038 | 0.5028 |
| MPL = 3 | 0.4828 | 0.5105 | 0.5136 | 0.5130 |
| MPL = 4 | 0.4895 | 0.5048 | 0.5128 | 0.5136 |
| MPL = 5 | 0.4894 | 0.4987 | 0.5097 | 0.5113 |
| MPL = 6 | 0.4864 | 0.5017 | 0.5151 | **0.5157** |

Table 13. Time slice accuracy for change point data of KD with $fp_2$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.5826 | 0.5827 | 0.5828 | 0.5828 |
| MPL = 2 | 0.6821 | **0.6824** | 0.6821 | 0.6816 |
| MPL = 3 | 0.6067 | 0.6456 | 0.6463 | 0.6467 |
| MPL = 4 | 0.6067 | 0.6446 | 0.6489 | 0.6494 |
| MPL = 5 | 0.6065 | 0.6448 | 0.6489 | 0.6501 |
| MPL = 6 | 0.6057 | 0.6481 | 0.6505 | 0.6530 |

Table 16. Class accuracy for change point data of KD with $fp_2$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.5235 | 0.5248 | 0.5253 | 0.5258 |
| MPL = 2 | **0.5774** | 0.5725 | 0.5766 | 0.5768 |
| MPL = 3 | 0.5346 | 0.5596 | 0.5641 | 0.5641 |
| MPL = 4 | 0.5368 | 0.5469 | 0.5503 | 0.5516 |
| MPL = 5 | 0.5364 | 0.5508 | 0.5566 | 0.5603 |
| MPL = 6 | 0.5255 | 0.5500 | 0.5594 | 0.5634 |

Table 17. Time slice accuracy for last data of KD with $fp_2$.

| timeslice | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | **0.9393** | 0.9391 | 0.9384 | 0.9376 |
| MPL = 2 | 0.9250 | 0.9256 | 0.9244 | 0.9230 |
| MPL = 3 | 0.9357 | 0.9325 | 0.9312 | 0.9300 |
| MPL = 4 | 0.9372 | 0.9345 | 0.9316 | 0.9303 |
| MPL = 5 | 0.9375 | 0.9364 | 0.9333 | 0.9269 |
| MPL = 6 | 0.9373 | 0.9378 | 0.9347 | 0.9329 |

Table 18. Class accuracy for last data of KD with $fp_1$.

| class | Delay=1 | Delay=2 | Delay=3 | Delay=4 |
|---|---|---|---|---|
| MPL = 1 | 0.7578 | **0.7676** | 0.7667 | 0.7647 |
| MPL = 2 | 0.7536 | 0.7602 | 0.7560 | 0.7554 |
| MPL = 3 | 0.7556 | 0.7557 | 0.7509 | 0.7549 |
| MPL = 4 | 0.7556 | 0.7556 | 0.7493 | 0.7526 |
| MPL = 5 | 0.7537 | 0.7519 | 0.7446 | 0.7425 |
| MPL = 6 | 0.7519 | 0.7533 | 0.7456 | 0.7467 |

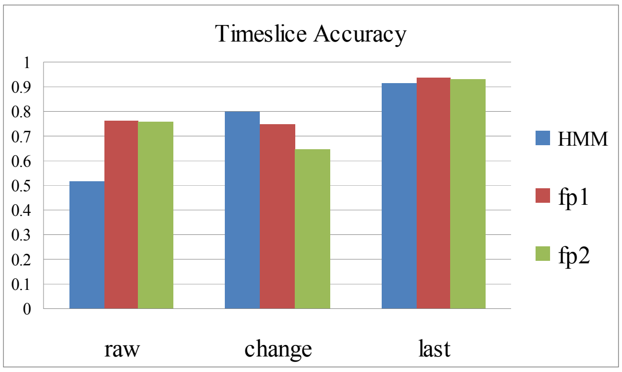


Figure 3. A comparison of time slice accuracy in KD.

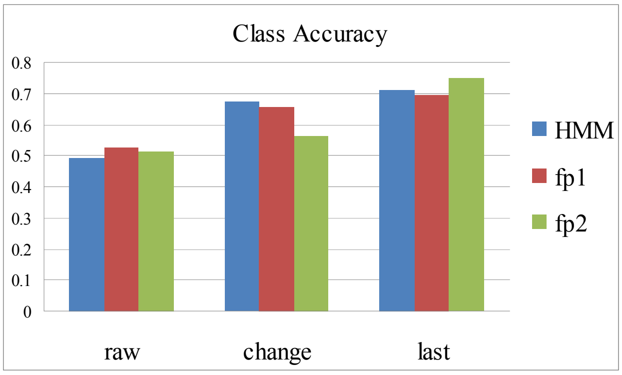


Figure 4. A comparison of class accuracy in KD.

## 5. Conclusion

In this paper, we propose a novel scheme based on minimal distinguishing patterns for activity recognition on multi-sensor data streams. First, the minimal distinguishing patterns are mined to determine the activity for each streaming sequence. Two estimation functions and the sequence classification algorithm are proposed to resolve the undecided cases effectively. As further work, the proposed scheme is easy to extend to online streaming data recognition and concept drift handling.

## Acknowledgments

This research was supported in part by the National Science Council of Taiwan, R. O. C. under contracts NSC100-2221-E-024-021 and NSC101-2221-E-024-026.

## References

[Chuzhanova 1988] N. A. Chuzhanova, Antonia J. Jones, and Steve Margetts, "Feature Selection for Genetic Sequence Classification," *Bioinformatics*, vol. 14, no. 2, pp. 139-143, 1998.

[Gopalratnam 2003] K. Gopalratnam and D. J. Cook, "Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction," in *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference*, St. Augustine, Florida, USA, 2003, pp. 38-42.

[Gopalratnam 2007] K. Gopalratnam and D. J. Cook, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm," *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 52-58, 2007.

[Hsu 2010] K. C. Hsu, Y. T. Chiang, G. Y. Lin, C. H. Lu, J. Y. J. Hsu, and L. C. Fu, "Strategies for Inference Mechanism of Conditional Random Fields for Multiple-Resident Activity Recognition in a Smart Home," in *Proceedings of the 23th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, Cordoba, Spain, 2010, pp. 417-426.

[Ji 2005] X. Ji, J. Bailey, and G. Dong, "Mining Minimal Distinguishing Subsequence Patterns with Gap Constraints," in *Proceedings of the 5th IEEE International Conference on Data Mining, Houston*, TX, USA, 2005, pp. 194-201.

[Kasteren 2008] T. V. Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate Activity Recognition in a Home Setting," in *Proceedings of 10th International Conference on Ubiquitous Computing*, New York, NY, USA, 2008, pp 1-9.

[Kaján 2006] L. Kaján, A. Kertést-Farkas, D. Franklin, N. Ivanova, A. Kocsor, and S. Pongor, "Application of a Simple Likelihood Ratio Approximant to Protein Sequence Classification," *Bioinformatics*, 22(23), 2006, pp. 2865-2869.

[Keogh 2000] E. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for data mining applications," in *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 285-289.

[Lesh 1999] N. Lesh, M. J. Zaki, and M. Ogihara, "Mining Features for Sequence Classification," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 1999, pp. 342-346.

[Lewis 1998] D. D. Lewis, "Naïve Bayes at Forty: The Independence Assumption in Information Retrieval," in *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany, 1998, pp. 4-15.

[Littlestone 1988] N. Littlestone, "Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm," *Machine Learning*, vol. 2, no. 8, 1988.

[Singla 2010] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Recognizing Independent and Joint Activities Among Multiple Residents in Smart Environments," *Journal of Ambient Intelligence and Humanized Computing*, Vol. 1, No. 1, 2010, pp. 57–63.

[Sonnenburg 2005] S. Sonnenburg, G. Rätsch, and B. Schölkopf, "Large Scale Genomic Sequence SVM Classifiers," in *Proceedings of the 22nd International Conference on Machine Learning*, New York, NY, USA, 2005, pp.848-855.

[WSU 2010] http://ailab.wsu.edu/casas/datasets/index.html

[Xing 2010] Z. Xing, J. Pei, and E. Keogh, "A Brief Survey on Sequence Classification," *ACM SIGKDD Explorations Newsletter*, vol. 12, no 1, pp. 40-48, June, 2010.

[Yakhnenko 2005] O. Yakhnenko, A. Sillvescu, and V. Honavar, "Discriminatively Trained Markov Model for Sequence Classification," in *Proceedings of the 5th IEEE International Conference on Data Mining, Houston*, Texas, USA, 2005, pp. 498-505.