Using Sensor Sequences for Activity Recognition by Mining and Multi-Class Adaboost

Md. Kamrul Hasan, Sungyoung Lee, and Young-Koo Lee
Department of Computer Engineering, Kyung Hee University, South Korea

Abstract - *In this paper, we present an activity recognition system using sensor sequence information generated from many binary on-off state sensors. When many sensors are deployed the number of sensor activation sequence or any combination will be huge and handling the combination is beyond the capability of traditional classifiers. We propose to mine important features from training data set and use multi-class Adaboost to further reduce the dimension. As a result we get a real-time lightweight classifier for activity recognition.*

Keywords: Activity Recognition, Maximum Likelihood, Mining, Adaboost, Sensor Sequence.

1 Introduction

Recognizing Activity of Daily Living (ADL) [1] is of particular interest to researchers for its various application domain, especially in healthcare industry. Detecting and learning the daily activities of elderly person can save the caregiver's time and give the elderly more independence.

ADLs involve the household devices and utensils. A user manipulates and uses the devices and utensils in a certain fashion. So, in typical setup, various types of simple sensors, especially binary on-off state sensors, are deployed in the environment. The sensors are supposed to be deployed and forgot. So, the system needs to gather training data over a period of time. Sensor activation sequences are then fed to the system to train and test the classifier. Due to unavailability of any special features, the sensor states of the whole house are usually taken as the observation at a time instance.

With many sensors, training data is not always enough to capture all the variations of an activity [2]. Moreover, there are noises in the data, because user's movements are not always purposeful, and even if purposeful, the movement can be for performing a different activity. So, we move away from using a high level model defining the networks for ADLs involving all possible combination of daily object usage. We rather focus on using the temporal relationships among the sensors. As for example, a 'preparing meal' activity is indicated by 'going to the kitchen', 'turning on the gas burner', and so on. Naturally, those sensors attached to the cooking utensils will be activated while the user is preparing a meal. Some other sensors can also be activated during that period as the user deviates from meal preparation. Finding the most relevant sensors and sequences and ignoring others, can contribute to the detection of the user activity.

However, capturing and calculating sequence information as features is huge and very computation intensive. Suppose, we want to calculated a sequence length of L and there are n sensors with s possible states, then there are $(ns)^L$ number of possible features that can be constructed. Given 100 sensors, and 5 sequence length, the number of possible activation sequence will be 100^5. Calculating such a huge number of features real-time and feeding to classifier is impossible. An alternative to calculating sensor sequences is to put the observation sequences to Hidden Markov Model (HMM)[3]. However, the key to the success of HMM is to provide longer sequence of observations which is unlikely to get in real-time activity recognition system. HMM uses conditional Probability of Activities, $P(Activity_1|Activiyt_2)$ and much of their results are dependent on the exact measurement of these probabilities. But user behavior usually varies person to person and even changes time to time. HMM suffers the same dimension problem for storing the probability tables. With n sensors each with s states, there are multiple of s^n observation probabilities, if all the sensors together represent the state of the environment. With 100 binary sensors we have multiple of 2^{100} observation probabilities, which cannot be loaded even on a 10GByte memory space. So, it is not lightweight.

We provide a real-time lightweight activity recognition system from sensor sequence information by reducing feature dimension using mining and Adaboost. We use four features: namely quantized maximum likelihood values for each activity, last sensor activated for each activity, sensor sequences and group of sensors activated. The number of sensor sequences and group of sensors activated can be huge. So, we first mine important sensor sequences and group of sensors activated from training data. Those features along with maximum likelihood and last sensor activated features are fed to Multi-class Adaboost to select most relevant features for classification. The reduced numbers of features are then used for recognition.

Adaboost makes a strong classifier based on weak hypotheses set each of which work just better than random. In case of ADLs, definite steps and models are not yet discovered and even if there is, there are variations in performing those steps. However, each ADL may have some indicative features which may not be unique to identify an activity but may help in classification. Based on this idea, we propose to find out important sensor sequences and group of sensors to be mined from training data. We also propose to use quantized maximum likelihood values and last sensor activated to be used. The features themselves are weak classifiers which are then fed to Adaboost. The features are called hypotheses. Mining is needed because there is a huge set of sensor sequence and group possible that without mining, the feature set to be dealt with will be huge.

Adaboost is suitable for situation when there is no key feature but the features together, may form a strong classifier. For recognizing ADLs the features we calculate are not key features rather are indicative of the activities being performed. So, Adaboost is the best choice for recognizing ADLs.

The original Adaboost algorithm was designed as binary classifier. Later it was extended to multi-class (actually multi-label) problem by putting multiple labels to a simple example. So, for multi-class problem, an example will retain the true label and in addition, will assume negative labels for all other classes (one-versus-all). Details can be found in [4] and an implementation has been provided named Adaboost.MH. However, a new algorithm, proposed by Zhu et al [5] looks almost same as original binary Adaboost algorithm and does not need to do the cumbersome process of Adaboost.MH. They have shown that a simple addition of a term *log(K-1)* with the update equation can convert it to multi-class. So, the output, instead of taking the sign, the class having the maximum value is chosen (Figure 1). An implementation has been provided by the name ICSIBoost.

Given: a set of training example $(x_1,c_1),\ldots\ldots,(x_m,c_m)$
Initialize observation weights $w_i=1/m$, i=1,2,…,m
for $t=1,\ldots,T$:
Train weak learning $L^t(x)$ using weights w_i
Get weak hypothesis h_t with error:

$$\varepsilon_t = \sum_{i=1}^{n} w_i \mathrm{I}\left(c_i \neq L^t(x_i)\right) / \sum_{i=1}^{n} w_i$$

Choose $\alpha_t = \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) + \log(K-1)$

Update:

$$w_i \leftarrow w_i . \exp\left(\alpha_t . \mathrm{I}\left(c_i \neq L^t(x_i)\right)\right)$$

for i=1,…,n
renormalize w_i
Output :

$$C(x) = \arg\max_k \left(\sum_{t=1}^{T} \alpha_t \mathrm{I}(L^t(x) = k\right)$$

Figure 1. Multi-class Adaboost

For real-time activity recognition we take 20 second data window from which we calculate the hypotheses selected by using mining and Adaboost during training. The number of hypothesis to be calculated is very less compared to huge set of possible features. That is why the algorithm is lightweight. The rest of the paper is organized as follows: Section 2 describes some related works, section 3 is describes our idea, section 4 is the result section and we conclude in section 5.

2 Related Works

Given a set of sensors deployed in the environment and with house hold utensils, can we develop a system that can recognize with the user/users is/are doing? This is the ideal setup for any activity recognition system, as it does not hamper privacy and also users are not burdened with carrying special devices. The work [6] that popularized the concept was done by Tapia et al in MIT. They deployed 77 binary on/off state sensors in a house and used a PDA to label the activities the person (subject 1) was doing. In another house they deployed 84 sensors for subject 2. However, the data collected from subject 1 has better quality in terms of labeling and noise. In our work , we used the data collected from subject 1 only.

Work [6] used different feature window for each activity, each window being the average of the activity. The feature window was shifted 3 minutes forward every time. Within the window, sensor activated (exists, E) and Before Time (BT) feature was calculated. The features were then fed to Multiclass Naïve Bayes classifier. The window size being very large, Before Time feature did not work better. Again as the number of features to be calculated for higher order sequences is huge, they did not go for that. The system with the 3 minutes sliding window, is not real-time.

The experience of the system deployment and pros-corns was shared in [1]. The system was further upgraded to incorporate RFID sensors, Tap sensors, Current sensors, Accelerometer sensors, motion sensors to find out which work better and the result was reported in [7]. It was found that motion sensor performs better. They also reported that RFID sensors are difficult to deploy with daily utensils and tags are not always read. They also suffered from lack of labeled data, as users were asked to live as naturally as possible. From video labeling 104 hours of data were collected.

The idea of deploying sensors in the environment was picked by Work [8]. They deployed 14 on/off sensors in the doors and recognized 7 ADLs only. They reported that change events are the most useful features. The event continues to be valid until another sensor event is fired. They used HMM and CRF for recognition. Change event is useful but reduces accuracy for activities, especially idle activity. HMM and CRF used conditional probabilities for activities, *P(Activity1|Activity2)* that is subject to change from user to user and time to time. Moreover with higher number of sensors, the classifiers fail to load the probability tables in memory which they failed to observe. They also reported that the classifiers provide highest accuracy when the data of the whole day is fed. That means the algorithm is not real-time.

Data mining has been proposed to find out periodic patterns for behavior learning [9]. The patterns are then used match similar pattern. Here time (1 hour) is used to restrict the searching window. In our work, we are using mining find patterns (may not be periodic) to recognize ADL, which is different than finding periodic patterns re-occurring. We use mining and Adaboost to reduce the feature dimension.

3 Main Idea

Sensors are activated and deactivated while the user is performing some ADL. If we take the samples of the sensor activations periodically, a sequence of activations can be obtained. The sequence of sensor activations may indicate what the user is actually doing. Based on this idea, we propose four types of features, namely: maximum likelihood, last sensor activated, sensor sequence, and sensor group. The features are calculated over a time window and fed to Multi-class Adaboost for classification (see Figure 2).

During the training, Adaboost uses a weak classifier in a series of rounds. For each round, a hypothesis is chosen and based on the results, a distribution of weights that indicate the importance of examples in the dataset for classification is updated. Importantly, the weights of each incorrectly classified example are increased so that the new classifier focuses more on those examples. Because the ADL problem under study, comprising a large number of sensors whose combined states may not have definite indicative features for activities, the Adaboost can be a good solution. The feature values from the four types of the features are the hypotheses in our case.

In later subsections from 3.1 to 3.4, we describe how the features are calculated. In section 3.5 we describe how mining and Adaboost reduce the feature dimension.

3.1 Maximum Likelihood Calculation

Suppose, we allow a maximum sequence of length T. So, a structure is constructed for each activity with the sensors activated in last T time slices. Each sensor activated at a time slice is a state of the structure. If no sensor is activated in a time slice, an idle sensor is assumed which favors all the activities equally. We assume first order markov property. So, states have transition with the previously activated states only. We omit the state transition links among the sensors in the same sample period because those sensors infer concurrency, not sequence.

Prior probability for each state in the structure is actually the prior probability of the respective sensor being activated during the activity. A state transition probability is the conditional probability of one sensor being activated given another sensor activated before within.

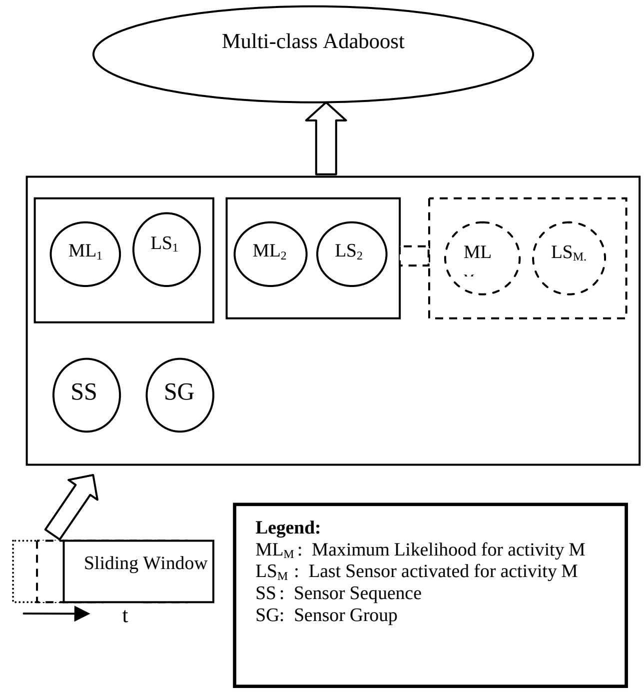

Figure 2. Classification Algorithm

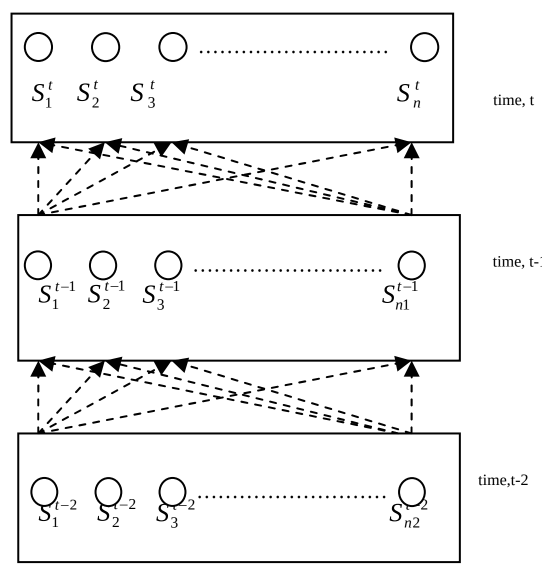

Figure 3. Structure created for each activity to calculate maximum likelihood

After constructing the structure for each activity, we calculate the likelihood for them. We use a maximum likelihood algorithm found in standard message passing literature [10]. Each state calculates the maximum likelihood using the formula below:

$$P_a(S_i^l) = P_a(S_i)P_a(O_{oi}) \quad \text{, l=last sequence}$$

$$P_a(S_i^l) = P_a(O_{oi})\max_c[P_a(S_i^l \mid S_c^{l-1})P_a(S_c^{l-1})] \quad \text{, otherwise} \quad \text{.... (1)}$$

Where,

$P_a(S_i)$ =Prior probability of sensor i for activity ‘a’

$P_a(S_j \mid S_i)$ = conditional probability of S_j given S_i for activity ‘a’

$P_a(O_{oi})$ =Probability of output O_o from sensor i for activity ‘a’

S_i^t = state constructed with sensor i at time t

$P_a(S_i^t)$ =Calculated probability (usually maximum likelihood) of S_i^t for activity ‘a’

$P_a(S_j^t \mid S_i^{t-1})$ =State transition probability of S_j^t given S_i^{t-1} $= P_a(S_j \mid S_i)$

The maximum likelihood values are then quantized into several ranges. When quantized values are used as hypothesis, each of them does not have much recognition power (may be better than random). The range values work as hypotheses in Adaboost.

3.2 Last Sensor Activated

Last sensor activated can provide a key event for some activities. For example, if the burner is on, it indicates cooking or if the shower is on, it indicates bathing. However, key sensors are not always activated. If sensors are placed in strategic places (knowing the key positions), it is possible to design a rule based activity recognition system using key sensor activation events. But in a randomly deployed sensor environment, it is not known whether there are any key sensors and whether they are on as long as the activities are being performed. That is why last sensor activated becomes a hypothesis in Adaboost, rather than becoming a key feature.

3.3 Sequence of sensors

The sequences are indicative of activity performed. For example, going to kitchen, turning on the burner can indicate preparing meal. However, many activities have common subsequences and hence sequences are not key features. So, sensor sequences need to be fed as hypotheses in Adaboost.

3.4 Sensor Group

Sensors activated during a period of time are indicative of activity being done. For example, refrigerator door open, kitchen cabinet open may indicate some meal preparation activity being performed. That is why; group of sensors activated is taken as hypotheses in Adaboost.

3.5 Mining and Adaboost for Feature Dimension Reduction

Mining is needed in the training phase of Adaboost. The possible number of sequence of sensors is exponential and is very difficult to handle, though not impossible. With 100 sensors there are 100^5 (10^{10}) sequences of 5-length possible. There are ${}^{100}C_5 + {}^{100}C_4 + {}^{100}C_3 + {}^{100}C_2$ (approx. 100^4 or 10^8) possibilities of grouping of sensors. Keeping track of every sequence is very costly. But many of those sequences or groups may not occur in training dataset. So, some mining technique should be used.

We use frequent item set mining algorithm proposed by Han et al [11]. The algorithm makes FP-Tree from a transactional database and then uses FP-growth algorithm to mine frequent patterns from the tree. The algorithm itself cannot mine sequences or groups of items from a set of transactions; rather it mines items from a single transaction. So, for mining sequence or group, we need to prepare the sensor data (see Table I for example data) as transactions and yet retaining the sequence information. If we allow a maximum window length of L, sensor activations of L sample windows should be merged together. To retain sequence information, sensors activated in a time slice are post fixed with a number in order of their arrival in the sequence. For example, sensors activated in the first time slice will all be post fixed with a number 1 before they are merged as a transaction (see table II). From sensor group mining, however, the post fix operation is not needed .

Table I: Sample training data for activity A1

Activity	Sensors Activated	Time
A1	a,b,c	00
A1	a,d	05
A1	e	10
....		
A1	a,b,f	100
A1	a	105
A1	e,f	115

Table II: Transactional database for sensor sequence mining for activity A1 with sequence length 3

Transaction#	Items
1	a1,b1,c1,a2,d2,e3
2	a1,d1,e2
3	e1
4	a1,b1,f1,a2,e3,f3
5	a1,e2,f2
6	e1,f1

The sensor sequence or sensor groups are supposed to be discriminating, so that they help Adaboost to decide the activity label. Here we adopt two rationales: The higher the support for a sequence or group within activity duration, the higher the probability that it is a key sequence or group to the activity. A sequence or group found in one activity and absent in another is likely to be a discriminating feature. We use mining to find out sequence or group of sensors occurring frequently within an activity. The sensor sequences thus mined for each activity are merged together sorted in separate feature buckets so that concurrently occurring sequences are put in different buckets. Same operation is done for sensor groups mined for all the activities individually. The task of finding discriminating sequences and groups is left to Adaboost.

The accuracy of Adaboost algorithm increases with more hypotheses. But the increment stops or the rate of increment reduces significantly after a while. The hypotheses set at that point are the reduced set of features needed. With Adaboost we could reduce the set of feature significantly, even after using the mined features.

4 Results

We are using the open data provided by MIT Place Lab [6,12]. They used 77 on/off state sensors in a one person apartment. The person lived there for two weeks and was given a PDA which provided periodic beeps to get input about what the user was doing. The method of labeling was named Experience Sampling Method (ESM) [6]. When the user used to do something, events were fired and the sensors sent the event to central server. The server kept the time stamped sensor event along with the labeling in a log. Often the user forgot to label the activities. Those periods were later presented to the user to label from their memory. Often the user forgot to turn off the sensors and those were added as noise to the data. Sometimes the user used to interleave between activities but only one label was taken. The number of example instances for each activity is small and contain noises.

We assumed 20 second time window with 5 second slide. So, a sample of sensor values is taken after each 5 seconds.

Quantization of Maximum Likelihood: We found that maximum likelihood values for each activity have some patterns and sometimes near to some values. So, I took only 200 value ranges for each activity. So, the total number of hypotheses possible is 19 X 200, as we dealt with 19 activity instances.

Frequent Itemset mining: The sensor sequence and group of sensors activated for each activity were mined from the whole dataset and then were merged together. The approach will mine frequent patterns for each activity, some of which may be important sequences. Moreover, the approach reduces number of items to be mined. Mining less supported items increase the hypotheses set but does not increase the recognition accuracy. As Adaboost selects the best hypotheses set to be used, less useful hypotheses are dropped automatically.

The set of sequences and groups that are possible, mutually exclusive groups are formed for different lengths. Suppose sequence ‘abc’ never occurred together with ‘cfg’. So, they are put in the same column as the feature to the adaboost. The input for the adaboost becomes something as shown below:

Class ML1 ML2 …ML19 LS1…LS19 SB1… SB2…GB1…GB2….

Where class is the class label, MLi is the Maximum Likelihood value for class i, LSi is the last sensor for class i, SBi is the i-th bucket for sensor sequence. GBj is the j-th bucket for sensor group.

We mined items with support 10 and around 1,40,000 items were mined. From that more than 60,000 beans were constructed. Then the input data was constructed to be fed to Adaboost.

Classification: We used SAMME algorithm for multi-class Adaboost proposed in [5]. Weak Threshold classifier was used to find out the hypotheses. For increasing number of iterations, this stopped increasing for iterations 20,000. That means 20,000 hypotheses need to be computed each time a decision is to be made by Adaboost.

Table III. Accuracy for different number of iterations in Adaboost

Iterations	5000	10000	15000	20000
Average	31.28	52.47	66.58	73.05

Table IV. Highest accuracy for iterations 20,000

Activity	Accuracy (%)
Bathing	78.43
Toileting	80.82
Going_out_to_work	62.18
Preparing_lunch	70.29
Preparing_dinner	48.80
Preparing_breakfast	69.44
Dressing	70.20
Grooming	84.45
Preparing_a_snack	79.27
Preparing_a_beverage	78.80
Washing_dishes	74.63
Doing_laundry	87.76
Cleaning	26.48
Putting_away_dishes	85.24
Washing_hands	73.84

Putting_away_groceries	80.87
Watching TV	57.52
Going_out_for_entertainment	84.36
Lawnwork	94.59
Average	73.05

Some activities like cleaning, watching TV have less number of actual activity instances in the dataset. Preparing dinner is much confused with preparing lunch and breakfast, which is natural. Hence the accuracies of these activities are less compared to others. Overall accuracy is still not quite perfect because of the small and noisy data set.

20,000 features extracted after using Adaboost is much less than around 1,40,000 features found by mining. But interestingly, both of then are far more smaller than over $(77+1)^5$ (one for idle sensor) sequential features possible. We did not find any other work, in activity recognition area to work with so huge feature set. We proposed mining with Adaboost for reducing features for dealing with such high dimensional feature set. With the reduced set of features the algorithm becomes lightweight and with 20 second window size, the algorithm is real-time.

5 Conclusion

We propose a real-time lightweight activity recognition algorithm. We propose four kinds of features. Two of those features involving sensor sequences may become huge for any conventional classification algorithm. So, we propose to use mining and Adaboost to reduce the dimensions of those features retaining the accuracy. So, the algorithm becomes lightweight. We also use small sliding window so that the recognition becomes real-time.

6 References

[1] ADL, Activity of Daily Living: http://en.wikipedia.org/wiki/Activities_of_daily_living.

[2] Intille, S.S., Larson, K., Tapia, E.M., Beaudin, J., Kaushik, P., Nawyn, J., and Rockinson, R., “Using a live-in laboratory for ubiquitous computing research”. In Proceedings of PERVASIVE 2006 , vol. LNCS 3968, K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Berlin Heidelberg: Springer-Verlag, 2006, pp. 349-365.

[3] Rabiner, L. R. and Juang, B. H, “An introduction to hidden Markov models”, IEEE ASSP Magazine: 4-15, Jan 1986

[4] Schapire, R. E. and Singer Y., “Improved Boosting Algorithms Using Confidence-rated Predictions”. Machine Learning, 37, 297–336 (1999), Kluwer Academic Publishers, Netherlands.

[5] Zhu, J., Zou, H., Rosset, S., and Hastie, T. , “Multi-class AdaBoost”. Statistics and Its Interface Volume 2 (2009) 349–360

[6] Tapia, E.M, Intille, S., and Larson, K., “Activity Recognition in the home setting using simple and ubiquitous sensors”. Proceedings of PERVASIVE 2004, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 158-175.

[7] Logan, B., et.al., “A Long-Term Evaluation of Sensing Modalities for Activity Recognition”. In the Proceedings of UbiComp 2007.

[8] Kasteren, T.V., Noulas, A., Englebienne, G., Krose, B., ”Accurate Activity Recognition in a Home Setting”. 10th International Conference on Ubiquitous Computing 2008, page(s): 1-9.

[9] Rashidi, R., Cook, D.J. , “Keeping the Resident in the Loop: Adapting the Smart Home to the User”. IEEE Transactions on Systems Man & Cybernetics, Part A., 2009

[10] Bishop, C.M, “Chapter 8: Graphical Models, Pattern Recognition and Machine Learning”, Springer.

[11] Han, J., Pei, J., Yin, Y., and Mao, R., “Mining frequent patterns without candidate generation”. Data Mining and Knowledge Discovery 8:53-87, 2004

[12] MIT Data , http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/