International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3525–3535

FEATURE SELECTIONS FOR HUMAN ACTIVITY RECOGNITION IN SMART HOME ENVIRONMENTS

Hongqing Fang[1], Raghavendiran Srinivasan[2] and Diane J. Cook[2]

[1]Department of Automation Engineering
College of Energy and Electrical Engineering
Hohai University
No. 8, Focheng West Road, Nanjing 211100, P. R. China
fanghongqing@sohu.com

[2]School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA, 99163, USA
raghav_raghu1110@yahoo.co.in; cook@eecs.wsu.edu

Received February 2011; revised August 2011

Abstract. *In this paper, three probabilistic models are applied to represent and recognize human activities from observed sensor sequences: Naïve Bayes classifier, forward procedure of a Hidden Markov Model and Viterbi algorithm based on a Hidden Markov Model. A variety of different feature selection methods is tested in order to reduce the dimensionality of the learning problem. The results show that the activity recognition performance measures of the three algorithms have a strong relationship with the dataset features that are utilized. Larger time feature values and smaller length size feature values will generate better results, relatively.*
Keywords: Activity recognition, Naïve Bayes classifier, Hidden Markov model, Viterbi algorithm, Smart home

1. **Introduction.** As the world's population ages, the number of elderly people living alone will enormously increase. With aging, health related problems also increase. Hence, the number of elderly people in need of assistance to have a normal life will increase rapidly. The number of care-givers will be outnumbered by the increasing elderly population. With the advent of smart home technologies, people with cognitive impairments can lead independent lives in their homes for longer duration of time. Smart homes can assist their residents by acting as a cognitive prosthesis, by handling various appliances/objects and also by facilitating emergency communication. Furthermore, cognitive health assessments performed in clinical settings do not always provide an adequate representation of a patient's behavior. Real life assessments of Activities of Daily Living (ADLs) can provide a better understanding of the subject than assessments performed in a clinical setting [1]. Computer vision sensing often works in the laboratory but fails in real home settings due to clutter, variable lighting, and highly varied activities. Video feeds have been used for activity recognition. Sensors such as microphones and cameras are commonly used as recording devices. Cameras and microphones face another challenge because they are perceived as invasive by most people [2-6]. Alternatively, motion sensor data can be used to recognize real-life activities done in a home. The CASAS smart homes [7] provide continuous monitoring capability that conventional methodologies lack. Being able to automate the activity recognition from human motion patterns, using unobtrusive sensors or other devices can be useful in monitoring older adults in their homes and keeping track of their ADLs and behavioral changes [8-15]. This could lead to a better understanding

of numerous medical conditions and treatments. The CASAS smart home project is a multi-disciplinary research project at Washington State University, USA, focused on the creation of an intelligent home environment. The approach is to view the smart home as an intelligent agent that perceives its environment through the use of sensors, and can act upon the environment through the use of actuators. The research goals of the CASAS smart home project are to enhance and improve quality of life, prolong stay at home with technology-enabled assistance, minimize the cost of maintaining the home and maximize the comfort of its inhabitants. In order to meet these goals, the smart home must be able to reason about and adapt to provide information. To implement the goal of the CASAS smart home project, a primary challenge is to design an algorithm that labels the activity performed by an inhabitant in a smart environment from the sensor data collected by the environment during the activity. Medical professionals also believe that one of the best ways to detect emerging medical conditions before they become serious is to look for changes in the ADLs. Recently, human activity discovery and recognition has gained a lot of interest due to its enormous potential in context aware computing systems, including smart home environments. To recognize residents' activities and their daily routines can greatly help in providing automation, security, and more importance in remote health monitoring of elder or people with disabilities. The main object of activity recognition in smart home environments is to find interesting patterns of behavior from sensor data and to recognize such patterns. Actually, the datasets include a large number of sensor events sequences generated by a various activities, and any activity annotated in dataset has various features. However, the influence of these feature values to activity recognition performance is seldom addressed, and it is necessary to research and test the relationship between feature selections and activity recognition accuracy rate.

2. **Smart Apartment Testbed and Data Collection.** The smart apartment testbed [12,13] for this study is located on Washington State University campus and is maintained as part of the ongoing CASAS smart home project. The smart apartment testbed includes three bedrooms, one bathroom, a kitchen, and a living/dining room. The smart apartment is equipped with motion sensors distributed approximately 1 meter apart throughout the space on the ceilings. Sensor data are captured using a sensor network that was designed in-house and is stored in an SQL database. The middleware uses a XMPP-based publish-subscribe protocol as a lightweight platform and language-independent method to push data to client tools with minimal overhead and maximal flexibility. To maintain privacy, participant names and identifying information are removed and encrypt collected data before it is transmitted over the network. After collecting data from the smart apartment testbed, the sensor events are annotated for ADLs, which are used to train the algorithms. A large number of sensor events are generated everyday. The data gathered by CASAS smart home is represented by the following parameters, which specify the number of features that are used to describe the sensor event. The default number of features is 5 and the default interpretation of the five features is shown below.

2.1. **Sensor ID.** This is an integer value in the range of 0 to the number of logical sensor values (defined in a separate parameter). The number of logical sensors can be any value between 1 and the number of physical sensors. The number of logical sensors is generally defined to be smaller than the number of physical sensors if multiple physical sensors will be mapped onto one logical sensor, thereby clustering the sensors together into one logical or functional unit. The ranges are not used by default.

2.2. **Time of day.** This is the input time of the sensor event but is discretized to an integer value (the discretization process is explained in a separate parameter entry). The default value is 5, which means the time ranges of one entire day are 0-5, 6-10, 11-15, 16-20, 21-24. The value of this feature is adjustable.

2.3. **Day of week.** The input date of the sensor event is converted into a value in the range of 0 to 6 that represents the day of the week on which the sensor event occurred.

2.4. **Previous activity.** This feature is an integer value that represents the activity that occurred before the current activity. The ranges are not used by default.

2.5. **Activity length.** This feature represents the length of the current activity measured in number of sensor events. The value of this feature is to calculate the value of length size threshold, and the default value is 3, which means the length size of each activity is distinguished by 3 thresholds: {small, medium, large}. The value of this feature is adjustable.

The generalized syntax of the dataset is given below:

Date Time Sensor ID Sensor Value ⟨label⟩

An example of the dataset of Night_wandering activity is:

```
{
  2009-06-10   03:20:59.08   M006   ON    Night_wandering   begin
  2009-06-10   03:25:19.05   M012   ON
  2009-06-10   03:25:19.08   M011   ON
  2009-06-10   03:25:24.05   M011   OFF
  2009-06-10   03:25:24.07   M012   OFF   Night_wandering   end
}
```

This example shows one sensors sequence corresponds to the Night_wandering activity with concrete Date, Time, Sensor ID, Sensor Value as well as activity label parameters.

3. Algorithms Applied for Activity Recognition.

3.1. **Naïve Bayes (NB) classifier.** An NB classifier uses the relative frequencies of feature values and the activity labels for the sample training data to learn a mapping from a data point description to a classification label. NB classifier determines activity labels probabilistically based on the number of sensor event of various kinds that occurred during the activity. In this research, all activities are represented by various features including the number of occurring times of SensorID, Time of day, Day of week, Previous activity and Activity length [12,15].

The activity label, A, is calculated as:

$$\arg\max_{a\in A} P(a\,|D\,) = \frac{P(D\,|a\,)\cdot P(a)}{P(D)} \tag{1}$$

In this equation, D represents the feature values. Since the denominator will be the same for all values of activity labeled a, it can be simply calculated only the numerator values, which means ignoring $P(D)$. $P(a)$ is estimated by the proportion of cases for which the activity label is a, and $P(D|a)$ is calculated as the probability of the feature value combination for the specific observed activity, i.e.,

$$P(D\,|a\,) = \prod_i P(d_i\,|a\,) \tag{2}$$

where i is the index of features.

3.2. **Hidden Markov model and Viterbi algorithm.** An Hidden Markov Model (HMM) is a statistical model in which the underlying model is a stochastic process that is not observable (i.e., hidden) and is assumed to be a Markov process which can be observed through another set of stochastic processes that produce the sequence of observed symbols [6,15,16]. The current state depends on a finite history of previous states. Actually, in this research, the current state depends only on the previous state. A hidden state is used to represent each of the separate activities. Each observable and hidden state is associated with a multidimensional probability distribution over a set of parameters. The parameters for the model are the feature values described in the previous section. Transitions between states are governed by transition probabilities. An HMM assigns probability values over a potentially infinite number of sequences. However, as the probabilities values must sum to one, the distribution described by the HMM is constrained.

For any given state, a sequence of observations can be generated according to the associated probability distribution. Since the goal is to recognize the activity that corresponds to a sequence of observed sensor events, one HMM is generated for each activity to be learned.

In an HMM, these hidden states are not directly visible, and are influenced by the observable states. Transition from any one state to another is governed by transition probabilities as in the Markov chain. Therefore, in a particular state an outcome can be generated according to the associated probability distribution.

The conditional probability distribution of any hidden state depends only on the value of the preceding hidden state. Similarly, the value of an observable state depends only on the value of the current hidden state.

Consider a system that has N distinct states, $\{s_1, s_2, \cdots, s_N\}$, and the actual state at time t is $q_t = s_i$, $1 \le i \le N$, then each state has M distinct observation symbols, which can be denoted as $\{v_1, v_2, \cdots, v_M\}$. In the theory of HMM, the observable variable $o_t = v_k$, $1 \le k \le M$ at time t depends only on the hidden state variable s_i at that time.

An HMM utilizes three probability distributions: the first is a probability distribution over initial states

$$\pi_i = P(q_1 = s_i) \tag{3}$$

Second, the state transition probability distribution represents the probability of transitioning from state i to state j, which has the form of

$$a_{ij} = P(q_t = s_j \,|q_{t-1} = s_i), \quad 1 \le i,\ j \le N \tag{4}$$

Third, the observation probability distribution indicates the probability that the state j would generate observation $o_t = v_k$

$$b_j(k) = P(o_t = v_k \,|q_t = s_j), \quad 1 \le j \le N,\ 1 \le k \le M \tag{5}$$

These distributions are estimated based on the relative frequencies of visited states and state transitions observed in the training data.

To calculate the probability of the observation sequence, $\{o_1, o_2, \cdots, o_T\}$, given a specific state q_t, i.e., $P(o_1, o_2, \cdots, o_t \,|q_t)$, where T is the number of observations in the sequence, the forward procedure can be applied. Consider the definition of a forward variable is

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t,\ q_t = s_i) \tag{6}$$

Equation (6) calculates the probability of the partial observation sequence, $\{o_1, o_2, \cdots, o_t\}$, until time t, and state s_i at time t. Actually, the forward variable is approximately equal to $P(o_1, o_2, \cdots, o_t \,|q_t)$, which ignores the probability of the partial observation sequence,

$P(o_1, o_2, \cdots, o_t)$. The reason for ignoring $P(o_1, o_2, \cdots, o_t)$ is similar to that of ignoring $P(D)$ in Equation (1). Then, the forward variable can be solved inductively as:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}\right] \cdot b_j(o_{t+1}) \tag{7}$$

The initialization is:

$$\alpha_1(i) = \pi_i \cdot b_i(o_1) \tag{8}$$

Once the forward procedure is done, the state q_t is known. Therefore, the forward procedure can be applied to recognize human activities with sensor event sequence of observations.

The aim of Viterbi algorithm is to find the single best state sequence, $\{q_1^*, q_2^*, \cdots, q_T^*\}$, for the given observation sequence, $\{o_1, o_2, \cdots, o_T\}$. The best score, i.e., the highest probability, along a single path at time t is define as:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1 q_2 \cdots q_t = s_i, o_1 o_2 \cdots o_t) \tag{9}$$

where $\delta_t(i)$ accounts for the first t observations and end in state s_i, and it can be solved inductively as:

$$\delta_{t+1}(j) = \left[\max_{1 \leq i \leq N} \delta_t(i) \cdot a_{ij}\right] \cdot b_j(o_{t+1}) \tag{10}$$

where $1 \leq j \leq N$ and $1 \leq t \leq T - 1$.

The initialization is:

$$\delta_1(i) = \pi_i \cdot b_i(o_1), \quad 1 \leq i \leq N \tag{11}$$

In each recursion of Equation (10), the label of a hidden state which Equation (9) is returned by

$$l_t^* = \arg\max_{1 \leq i \leq N} [\delta_t(i)] \tag{12}$$

Once the procedure is done, the best hidden state label sequence can be obtained as $\{l_1^*, l_2^*, \cdots, l_T^*\}$, which corresponds to the best hidden state sequence $\{q_1^*, q_2^*, \cdots, q_T^*\}$.

To implement the forward procedure of an HMM, as well as Viterbi algorithm, every activity is treated as a hidden state. If a total of N activities are labeled in the dataset to be recognized, then, HMM and Viterbi algorithms include N hidden states. Each hidden state denotes one of the N modeled activities. Next, each sensor is treated as an observable state, because of each used sensor is observable in the dataset.

The training data are used to learn the transition probabilities between states for the corresponding activity model and to learn probability distributions for the feature values of each state in the model. For this, the prior probability (i.e., the start probability) of every state can be calculated based on the collected data. The prior probability represents the belief about which state HMM is in when the first sensor event is seen. For a state (i.e., activity) a, this is calculated as the ratio of instances for which the activity label is a. The transition probability which represents the change of the state in the underlying Markov model, can also be calculated. For any two states a and b, the probability of transitioning from state a to state b is calculated as the ratio of instances having activity label a followed activity label b, to the total number of instances. The transition probability signifies the likelihood of transitioning from a given state to any other state in the model and captures the temporal relationship between the states.

Furthermore, the emission probability represents the likelihood of observing a particular sensor event for a given activity. This is calculated by finding the frequency of every sensor event as observed for each activity.

4. Tests Results.

4.1. **Training activities.** A total of 10 activities were performed in the CASAS smart apartment by 2 volunteers to provide physical training data for the three algorithms. These activities include both basic and more complex ADLs that are found in clinical questionnaires. These activities are: Bed_to_toilet (activity 0, A0), Breakfast (activity 1, A1), Bed (activity 2, A2), Computer_work (activity 3, A3), Dinner (activity 4, A4), Laundry (activity 5, A5), Leave_home (activity 6, A6), Lunch (activity 7, A7), Night_wandering (activity 8, A8) and Resident_medicine (activity 9, A9). The data have been collected in the CASAS smart apartment testbed for 55 days, which resulting in total 600 instances of theses activities and 647,485 collected motion sensor events.

4.2. **Time feature selections.** In this case, the length feature value is defined as the default value 3. This means that three activity length ranges are used. However, the time feature values are compared for different numbers of ranges including 1, 2, 3, 4, 5, 6, 8, 12 and 24, respectively. The 3-fold cross validation is applied in the data.

Table 1. Comparison results for total activities recognition accuracy rate

Algorithms	1	2	3	4	5	6	8	12	24
NB	0.685	0.723	0.732	0.755	0.788	0.775	0.775	0.803	**0.815**
HMM	0.550	0.577	0.613	0.628	0.683	0.687	0.690	0.717	**0.725**
Viterbi	0.567	0.590	0.613	0.617	0.687	0.692	0.690	0.715	**0.735**

From Table 1, it can be found that with the increasing number of time feature values, the trends of the total activities recognition accuracy rate increase for HMM and Viterbi algorithms, respectively. Whereas, the results for NB are slightly different. However, the results of all the three algorithms show that the highest total activities recognition accuracy rate can be obtained when time feature value is defined as 24. Further, NB outperforms other algorithms significantly with each tested time feature value. However, the comparison results of HMM and Viterbi algorithms are slightly different.

Table 2 shows that NB has the best recognition accuracy rate when time feature value is defined as 24 for activities 0, 1, 2, 3, as well as 8, the rate is 50% of all activities. When the time feature value is 12, the best results of activity 4 and 7 are generated by NB, with a proportion of 20%. Similarly, HMM generates the best recognition accuracy rate for activities 0, 1, 3, 4, and 8 when the time feature value is 24. Also, HMM has the best recognition accuracy rate for activity 6 and 9 when the time feature value is 12. Again, Viterbi algorithm has the best recognition accuracy rate for activities 0, 2, 3, 4, and 8 when the time feature value is 24, and it has the best recognition accuracy rate for activities 6 and 9 when the time feature value is 12. Therefore, it can be concluded that a relatively larger time feature value will result in a better recognition accuracy rate of a specific activity by NB, HMM and Viterbi algorithms. Moreover, NB outperforms other algorithms for activities 1, 3, 4, 6, 7 and 8. Viterbi algorithm outperforms other algorithms for activity 2. HMM and Viterbi algorithms have the same best results than NB for activity 0. All of the three algorithms have the same best results for activity 9.

One important point to be noted is that a lot of optimal results shown in Tables 1 and 2 are not generated under only one specific time feature value (i.e., the algorithms generate same optimal results under different time feature values). However, in these tests, only the maximal optimal time feature value for each activity is listed above. Even though, it shows that most of the activities have better results with a higher time feature value for all of the three algorithms applied. The reasons can be explained from the statistical data, which shows the hour-by-hour sensors events proportion of the 10 activities. Since one day has 24 hours, therefore, if the time feature value is defined as 24, which means 24 separate time zones are defined, hour-by-hour. Actually, the living habits or ADLs have strong relationship with time of the residents in CASAS smart home, e.g., for activity 0 (bed-to-toilet), 17.75% sensors events occur in the time zone of (0:00-1:00), 29.12% sensors events occur in the time zone of (2:00-3:00), and there are no sensors events occur in the time zone of (8:00-22:00). Larger time feature value means more precise time zone resolution, which generates relatively better results.

Table 2. Comparison results for each activity recognition accuracy rate

Activities	Algorithms	1	2	3	4	5	6	8	12	24
	NB	0.233	0.233	0.233	0.200	0.200	0.233	0.233	0.300	**0.333**
0	HMM	0.300	0.333	0.333	0.333	0.400	0.400	0.466	0.333	**0.500**
	Viterbi	0.300	0.300	0.300	0.333	0.400	0.400	0.467	0.400	**0.500**
	NB	0.896	**0.917**	0.875	0.896	**0.917**	0.875	0.875	**0.917**	**0.917**
1	HMM	0.271	0.271	0.771	0.354	0.563	0.771	**0.833**	0.813	**0.833**
	Viterbi	0.250	0.271	0.750	0.333	0.563	0.750	**0.813**	0.792	0.792
	NB	0.696	0.700	0.715	0.768	0.831	0.812	0.812	0.826	**0.860**
2	HMM	0.700	0.700	0.696	0.787	**0.855**	0.821	0.773	0.845	0.845
	Viterbi	0.749	0.734	0.720	0.783	**0.884**	0.865	0.797	0.870	**0.884**
	NB	**0.500**	0.478	0.478	0.478	0.478	0.457	0.457	0.478	**0.500**
3	HMM	0.217	0.239	**0.261**	**0.261**	0.217	**0.261**	**0.261**	**0.261**	**0.261**
	Viterbi	0.217	0.239	0.217	**0.261**	0.217	**0.261**	**0.261**	**0.261**	**0.261**
	NB	0.548	0.786	0.810	0.929	0.810	0.810	0.810	**1.000**	0.976
4	HMM	0.833	0.833	0.619	0.857	0.738	0.690	0.857	0.905	**0.929**
	Viterbi	0.857	0.833	0.619	0.833	0.762	0.738	0.833	0.857	**0.905**
	NB	**0.4**	0.3	**0.4**	0.3	**0.4**	0.3	0.3	0.3	0.3
5	HMM	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	0.4	0.2	0.4
	Viterbi	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	0.4	0.2	0.4
	NB	0.928	0.928	**0.942**	0.928	0.928	**0.942**	**0.942**	0.928	0.928
6	HMM	0.826	0.855	0.855	0.841	0.870	0.841	0.870	**0.884**	0.826
	Viterbi	0.841	0.855	0.826	0.855	0.855	0.841	**0.870**	**0.870**	0.826
	NB	0.486	0.784	0.838	0.784	0.892	0.838	0.838	**0.946**	0.892
7	HMM	0.216	0.541	**0.730**	0.541	0.595	0.703	0.541	0.622	0.622
	Viterbi	0.243	0.541	**0.703**	0.514	0.595	0.676	0.568	0.595	0.649
	NB	0.821	0.821	0.821	0.851	**0.940**	**0.940**	**0.940**	0.910	**0.940**
8	HMM	0.313	0.299	0.313	0.448	0.553	0.507	0.552	0.567	**0.597**
	Viterbi	0.284	0.299	0.299	0.373	0.493	0.463	0.493	0.522	**0.582**
	NB	0.682	**0.727**	0.705	0.705	0.705	0.705	0.705	0.705	0.682
9	HMM	0.614	0.636	0.614	0.591	0.659	0.659	0.705	**0.727**	0.682
	Viterbi	0.614	0.682	0.682	0.614	0.659	0.682	**0.727**	**0.727**	0.705

4.3. **Length feature selections.** In this case, the time feature value is defined with a larger value as 24, and the length feature value is defined from 2 to 45, respectively.

Figure 1 shows the comparison results of the trends of total activities recognition accuracy rate generated by NB, HMM and Viterbi algorithms with the increasing of length feature value. It can be found that in this test, NB outperforms HMM and Viterbi algorithms, significantly. The optimal value for the length feature for NB is 5. HMM and Viterbi algorithms have the same optimal results with length feature value of 8, and the

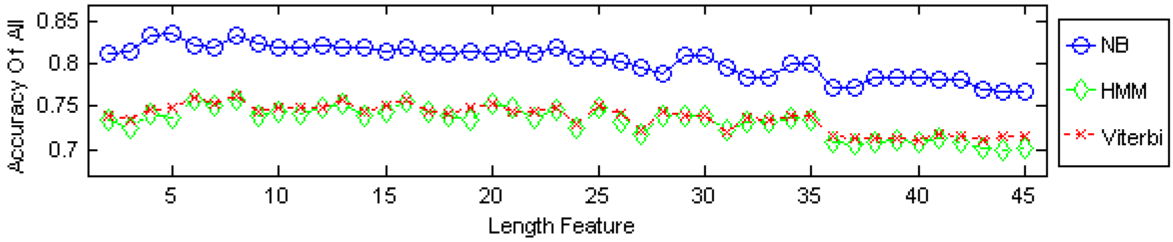

FIGURE 1. Comparison results of the trends of total activities recognition accuracy rate

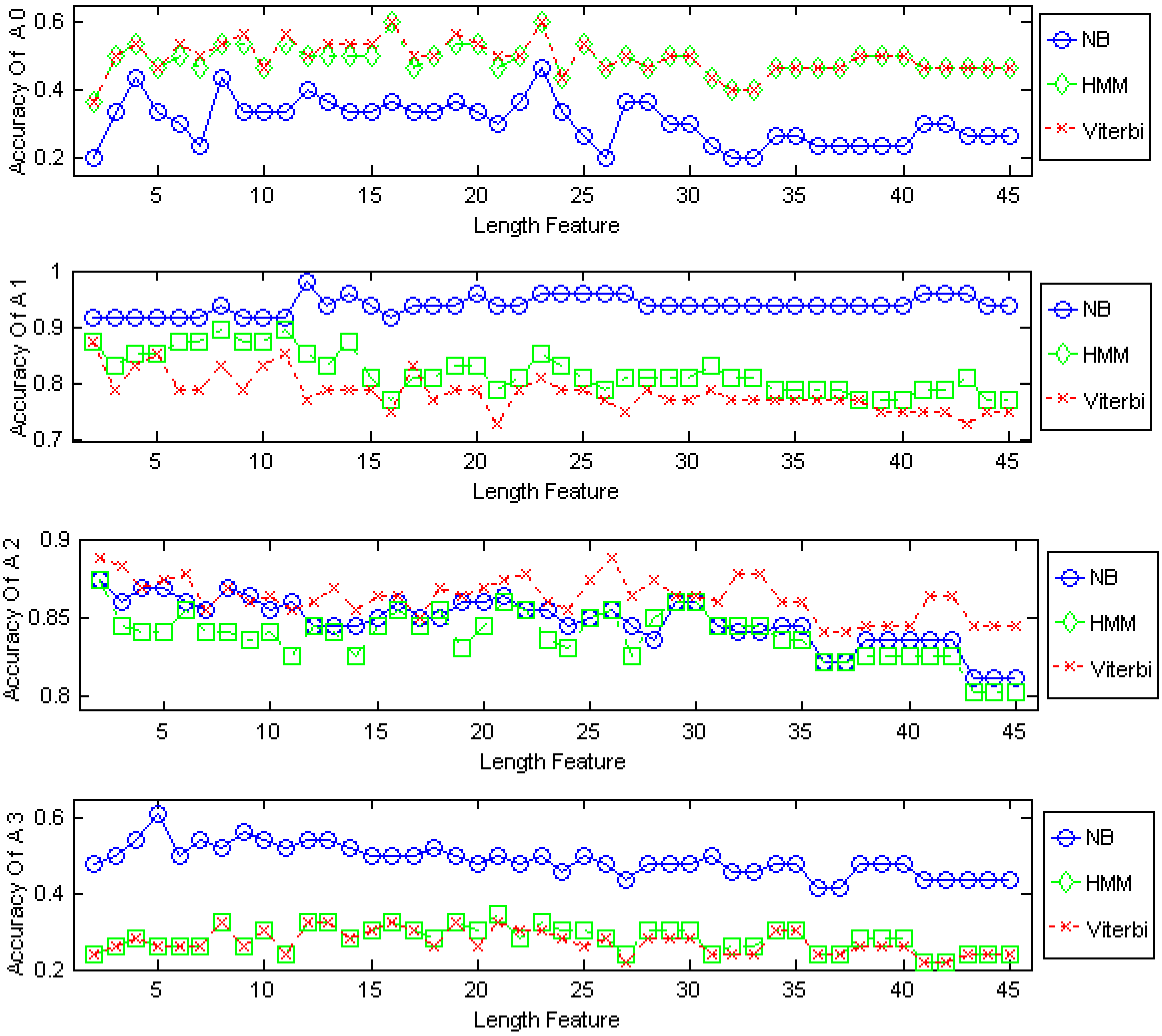

FIGURE 2. Comparison results of the trends of recognition accuracy rate for activities 0-3

other results of both algorithms are similar. HMM also has another optimal result with the length feature value of 6. Moreover, it shows that with the increasing of the length feature value, the results of all of the three algorithms appear oscillations and even become worse.

Figures 2 and 3 compare the results of the accuracy recognition trends of activities 0 to 9 generated by NB, HMM and Viterbi algorithms with increasing length feature values. It can be seen that NB significantly outperforms HMM and Viterbi algorithms for activities

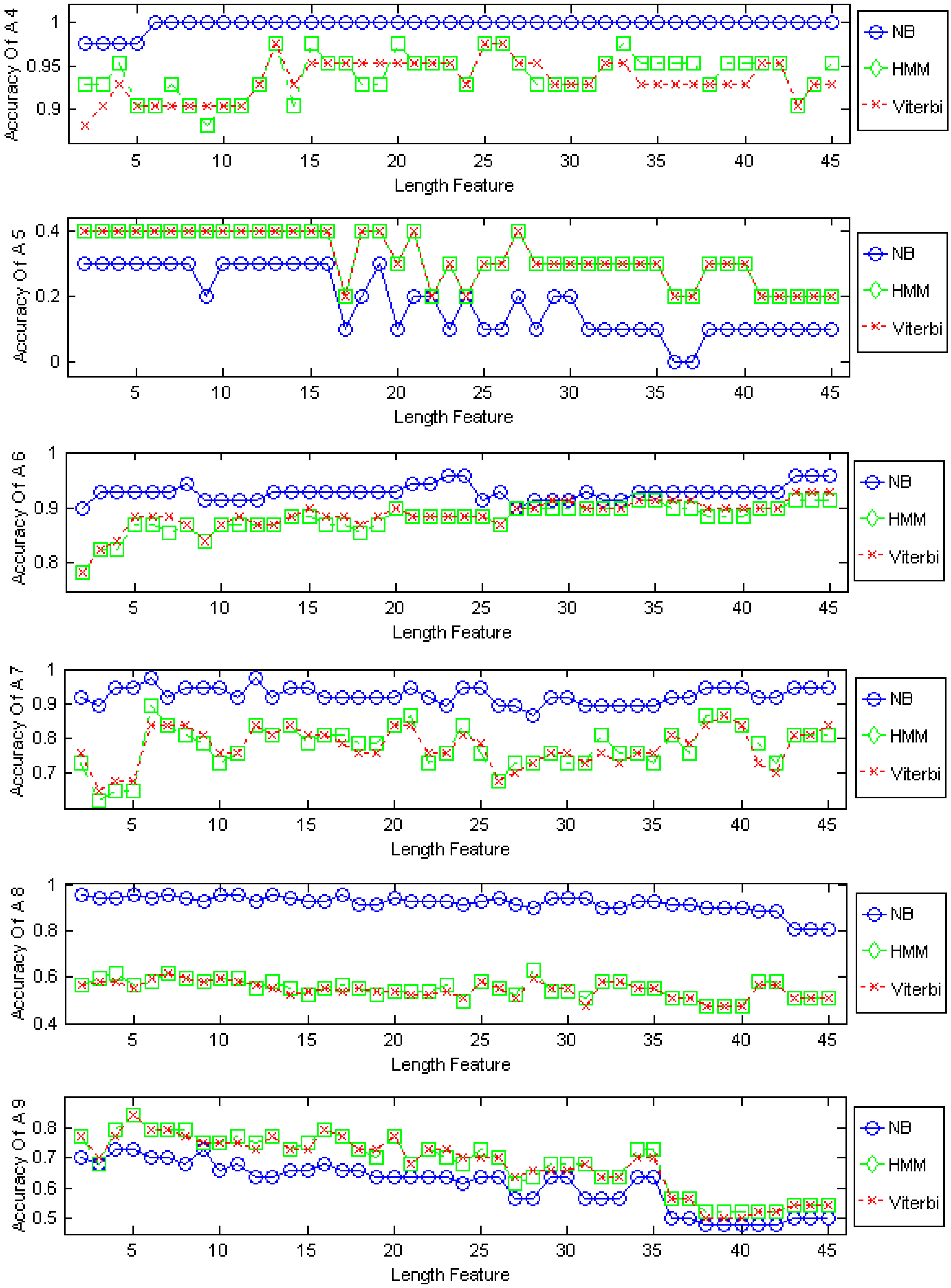

FIGURE 3. Comparison results of the trends of recognition accuracy rate for activities 4-9

1, 3, 4, 6, 7 and 8. Viterbi algorithm outperforms NB and HMM algorithms for activity 2. Both HMM and Viterbi algorithms have the same best results which are better than NB for activities 0, 5, and 9. It also can be seen that the results of HMM and Viterbi algorithms are close and usually the same, especially for activities 0, 3, 5, 6, 7, 8 and 9. The accuracy rates still show oscillations with increasing length feature values.

Again, it should be noted that the optimal results shown in Figures 2 and 3 are not generated by only one specific length feature value, i.e., the algorithms generate same optimal results under different length feature values. Actually, only the minimum optimal length feature value for each activity is given in this discussion. However, the results show that, it is better to define a small length feature value to get a better result. The reasons can be explained from the statistical data of sensors events length size of each activity. Different activity has different statistical data of sensors event length size; e.g., activity 6 has an average sensors event length size of 6, and in contrast, activity 4 has an average sensors event length size of 534. Larger length feature value will result in smaller length threshold value, which will generate more length features. Since the probability of feature given a specific activity is the product of the probabilities of each sub-feature given this activity, as shown in Equation (2), more length features will generate a smaller probability of feature given this activity. Therefore, a larger length feature value even generates relatively worse results.

5. **Conclusions.** This paper applies three probabilistic models to represent and recognize activities based on observed sensor sequences. In order to optimize these algorithms, the relationship between feature selections and recognition accuracy has been tested. From the results, it can be concluded that the activity recognition performances of the three algorithms have strong relationships with the feature values of datasets. Larger time feature values and smaller length feature values will generate better results, relatively. However, for different activities, the recognition accuracy rate results vary. In addition, this research is useful to select feature values to generate relatively better recognition accuracy for the three algorithms. In future work, the methods of automatically selecting feature values will be studied.

Acknowledgment. This work is partially supported by the Fundamental Research Funds for the Central Universities, China, under Grant No. 2009B01114, Qing Lan Project, Jiangsu Province, China, and the National Natural Science Foundation of China under Grant No. 60874013.

REFERENCES

[1] T. Barger, D. Brown and M. Alwan, Health status monitoring through analysis of behavioral patterns, *IEEE Trans. Syst. Man. Cybern. A Syst. Hum.*, vol.35, no.1, pp.22-27, 2005.

[2] U. Maurer, A. Smailagic, D. Siewiorek and M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, *Proc. of the International Workshop on Wearable and Implantable Body Sensor Networks*, Cambridge, Massachusetts, pp.113-116, 2006.

[3] O. Brdiczka, P. Reignier and J. Crowley, Detecting individual activities from video in a smart home. *Lect. Notes Comput. Sci.*, vol.4692, pp.363-370, 2007.

[4] C. Tsai, K. Cho, W. Yang, Y. Su, C. Yang and M. Chiang, A support vector machine based dynamic classifier for face recognition, *International Journal of Innovative Computing, Information and Control*, vol.7, no.6, pp.3437-3455, 2011.

[5] J. Lee and S. Lin, Hierarchical face recognition scheme, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5439-5450, 2010.

[6] Q. Zhang, C. Zhou and J. Zhao, Face recognition based on FLDA, CPCA and improved HMM, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.801-807, 2010.

[7] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders and B. Thomas, Collecting and disseminating smart home sensor data in the CASAS project, *Proc. of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, Boston, 2009.

[8] D. Cook, Health monitoring and assistance to support aging in place, *J. Univers. Comput. Sci.*, vol.12, no.1, pp.15-29, 2006.

[9] C. Wren and E. Munguia-Tapia, Toward scalable activity recognition for sensor networks, *Lect. Notes Comput. Sci.*, vol.3987, pp.168-185, 2006.

[10] J. Yin, Q. Yang, D. Shen and Z. Li, Activity recognition via user-trace segmentation, *ACM Trans. Sensor Netw.*, vol.4, no.4, 2008.

[11] C. Stauffer and W. Grimson, Learning patterns of activity using real-time tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, no.8, pp.747-757, 2000.

[12] A. Crandall and D. Cook, Coping with multiple residents in a smart environment, *J. Ambient Intell. Smart Environ.*, vol.1, no.4, pp.1-12, 2009.

[13] P. Rashidi and D. Cook, Keeping the resident in the loop: Adapting the smart home to the user, *IEEE Trans. Syst. Man. Cybern. A Syst. Hum.*, vol.39, no.5, pp.949-959, 2009.

[14] E. Kim, S. Helal and D. Cook, Human activity recognition and pattern discovery, *IEEE Pervasive Comput.*, vol.9, no.1, pp.48-53, 2010.

[15] G. Singla, D. Cook and M. Schmitter-Edgecombe, Tracking activities in complex settings using smart environment technologies, *Int. J. Biosci. Psychiatr. Technol.*, vol.1, no.1, pp.25-35, 2009.

[16] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. of IEEE*, vol.77, no.2, pp.158-175, 1989.