Optimal Number of States in Hidden Markov Models and its Application to the Detection of Human Movement

Satyam Jeebun*
Department of Electrical and Electronic Engineering
University of Mauritius
Réduit
E-mail : satyam.jeebun@gmail.com

Rajeshree Ramjug-Ballgobin
Department of Electrical and Electronic Engineering
University of Mauritius
Réduit
E-mail: r.ramjug@uom.ac.mu

Tarik Al-ani
ESIEE-PARIS
Département Informatique
France
E-mail : t.alani@esiee.fr

Paper accepted on 13 August 2015

Abstract

In this paper, Hidden Markov Model is applied to model human movements as to facilitate an automatic detection of the same. A number of activities were simulated with the help of two persons. The four movements considered are walking, sitting down-getting up, fall while walking and fall while standing. The data is acquired using a biaxial accelerometer attached to the person's body. Data of the four body gestures were then trained to construct several Hidden Markov models for the two people. The problem is to get a good representation of the data in terms of the number of states of the HMM. Standard general methods used for training pose some drawbacks i.e. the computational burden and initialisation process for the model estimate. For this reason, a sequential pruning strategy is implemented to address the problems mentioned.

Keywords: Hidden Markov Models, sequential pruning strategy, Bayesian Inference Criterion

**For correspondences and reprints*

1. INTRODUCTION

Hidden Markov model (HMM) has been very popular for modelling sequence of data. HMMs were successfully implemented in automatic speech recognition (R.Rabiner, 1989), handwriting recognition (Hu, et al., 1996), gesture recognition (Al-Ani, et al., 2007) (Ślusarczyk & Augustyniak, 2010), bioinformatics (Churbanov & Winters-Hilt, 2008), DNA and protein analysis (Burge & Karlin, 1997) (Peshkin & Gelfand, 1999) and amongst others. In this work, Hidden Markov model is exploited for gesture recognition. Among the rudimentary human movements like walking and sitting, fall also forms part of our life. However, fall within the elderly is very problematic. Many of our elders suffer a lot of pain if ever they collapse to the ground. Fall can lead to disastrous health condition or even death amongst elderly people. Studies have found that fall is ranked as the sixth leading cause of death among people over 65 years of age. And more than 90% of hip fractures occur as a result of fall after the age of 70 (Colorado State University, 2012) and (M.Tineti, et al., 1994). Also, 47% of the old persons who have fallen do not suffer from injuries but unfortunately cannot get up without assistance (Learn Not To Fall, 2012). Furthermore, a lot of the people who had a fall develop a fear of falling for a second time. This results in a limitation of their daily activities and this leads to lesser mobility and loss of physical fitness (Centers for Disease Control and Prevention, 2012). Like anyone of us, many of our elders prefer to stay at home instead in a care centre. And many of them stay alone at home. Incidents like a fall can happen any time. Within minutes after the incident proper care must be given to them. In this context, there need to be a system that automatically initiates for emergency unit. To resolve this problem; there need to have continuous monitoring between a healthcare provider and the patient who lives in his house. If ever a fall occurs the automatic system would initiate help. For automatic fall recognition, artificial intelligence approaches such as HMM (Ślusarczyk & Augustyniak, 2010) and (Al-Ani, et al., 2007) or Artificial Neural Networks (Zheng & Koenig, n.d.) can be used. The work from paper of (Al-Ani, et al., 2007) is being re-evaluated here to improve the results the authors arrived at. This is an HMM based approach for recognizing human activities (walking, sitting down-getting up, fall while walking and fall from an upright position). For each activity, we have to construct an HMM. For this, the observed data needs to be trained so that it can be represented by a model with an optimum number of states. A sequential pruning strategy (Bicego, et al., 2002) is moreover applied to get a better estimate for each model after training the HMMs. The data were captured using a biaxial accelerometer ADXL202E (Analog Devices, n.d.). The accelerometer is mounted on a microcontroller (Microchip, n.d.). Both devices are fastened to the waist of a person and the apparatus is allowed to capture data in real time. (Figure 1) The device picks up horizontal and vertical accelerations of the subject while he is moving.

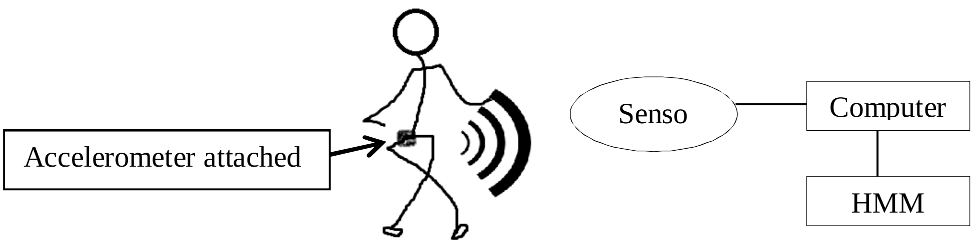

Figure 1 – Experimental Setup for data capture

The paper is organized as follows: Section II presents a concise description of HMMs, section III introduces the sequential pruning strategy for HMM training and the model selection criterion. The results are shown in section IV and section V concludes the paper along with some future works.

2. HIDDEN MARKOV MODEL

A Hidden Markov Model (R.Rabiner, 1989) is a stochastic finite state machine where we can only observe a sequence of output, $\boldsymbol{O} = o_1,\ o_2, \dots o_T$. These observations are generated by an underlying random state sequence, $\boldsymbol{Q} = q_1,\ q_2, \dots q_T$ and is assumed to be a first-order Markov process since the next state depends on the previous state only. Another important assumption while using HMM is that the probabilities (whether transition, emission or initial state probabilities) do not alter in time i.e. they are time independent. When dealing with HMM, two main structures are commonly employed i.e. the ergodic model and left-right model. The ergodic model is one in which all the states are fully

connected wherein any state can reach any other state with only one transition. For a first order Markov process with N states, there are N^2 transitions in an ergodic structure. The example in Figure 2 shows the structure of an ergodic model. As for the left-right model, there is no possibility of transitioning from a current state to a previous one. Figure 3 shows the structure of a left-right model of an HMM. Thus, the HMM can be represented by a graph of N states that can emit either discrete information or a continuous data derived from a Probability Density Function (PDF)

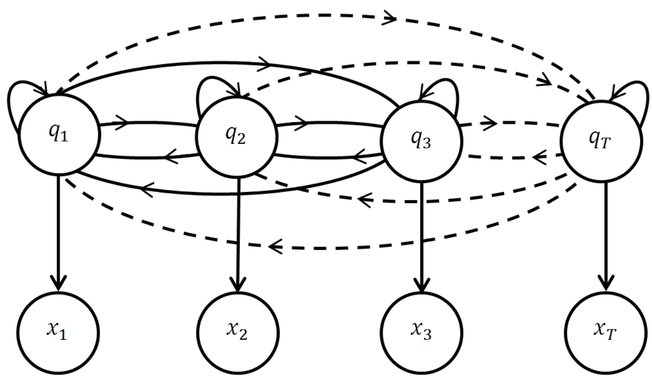

Figure 2: Structure of an ergodic HMM

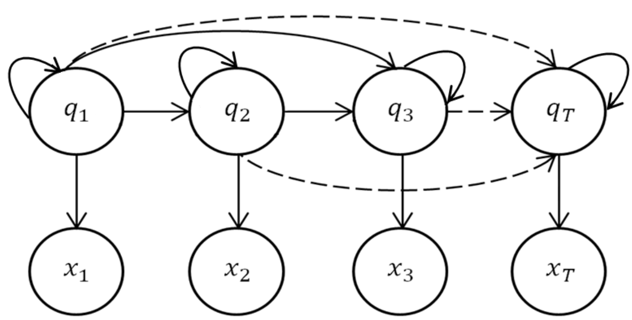

Figure 3: Structure of a left-right HMM

A discrete HMM, λ is characterized by the following:

- A set of hidden state

$$S = \{s_1, \ s_2, \dots s_k\} \quad \text{with} \quad \vec{\boldsymbol{Q}} \in \boldsymbol{S^k}.$$

- A vector of the initial state probability,

$$\boldsymbol{\pi} = P(s_i) = P(q_1 = s_i); 1 \leq i \leq k \tag{1}$$

where $s_i \in S$; $\sum_{s_i \in \boldsymbol{S}}^{k} \pi = 1$.

- The state transition matrix $\boldsymbol{A}$ (dimension $k \times k$) is the probability of going from state s_i to state s_j.

$A_{ij} = P(s_j | s_i)$

$$A_{ij} = P\left(q_{t+1} = s_j \middle| q_t = s_i\right); 1 \leq i, \mathrm{j} \leq k \tag{2}$$

where $s_i, s_j \in \mathrm{S}$; $\sum_{s_j \in \boldsymbol{S}}^{k} A_{ij} = 1$.

A_{ij} is named a stochastic matrix and we shall consider only stationary matrix i.e. the transition matrix does not depend on time *t*.

- A set of observation symbol

$X = \{x_1, \ x_2, \dots x_m\}$.

- An emission probability matrix, $\boldsymbol{B}$ (dimension $k \times m$) is the probability of an output symbol x_j from state s_i i.e.

$$\boldsymbol{B} = b(x_j|s_i) = P(o_t = x_j|q_t = s_i) \quad 1 \le i \le k; 1 \le j \le m \tag{3}$$

where $\sum_{j=1}^{m} b(x_j|s_i) = 1$.

The Hidden Markov Model can be summarised by a system comprising of two set of states, the hidden states and observable states and three sets of probabilities. Also the observable sequence is related to an underlying Markov process by the use of probability theory. Thus a discrete HMM is specified by the tuple **λ**= (**A, B, π**). For a continuous HMM, the only parameter **B** is taken to follow a normal distribution $X \sim N(\mu, \sigma^2)$ whereby parameters μ and σ^2 are the mean and variance (used as a covariance matrix, Σ) of the continuous random observed data respectively. All other parameters for a continuous HMM are same as that of the discrete HMM.

2.1 An HMM is solved by working out these three problems:

i. The evaluation step: Given the observation sequence O, and a model λ, how do we efficiently compute the probability of the observation sequence given the model. (i.e. likelihood) This is solved using the Forward Algorithm. (Baum, 1972)

ii. The decoding step: Finding the most probable or optimum state sequence that generated the observed data $argmax_q\ P(S|O,\lambda)$. This is solved using the Viterbi algorithm (Forney, 1973).

iii. Training step: Given the observation sequence O, we want to find the HMM parameters $\hat{\lambda}$ which maximise $P(O|\lambda)$ i.e. $\hat{\lambda} = argmax_\lambda\ P(O|\lambda)$. For HMM training, the Baum-Welch algorithm or the Forward-Backward algorithm is used. (Baum, et al., 1970)

For the evaluation problem, suppose we have a set of HMMs describing distinctive systems (models) and we have an observation sequence. We would like to know which model from the sets of HMMs, have the observed sequence been most probably be generated. That is how likely the data is originated from a particular model. The Forward algorithm recursively computes the probability of a model given an observation sequence.

The problem when dealing with HMM is to learn the parameters of the model which depends on unobserved latent variables. Hence, these parameters of interest need to be inferred in a way that it corresponds as close to reality. The Expectation-maximisation (EM) algorithm (Dempster, et al., 1977) is a well-known statistical algorithm that solves for this inference problem through an iterative process. This algorithm works by executing the expectation step (E step) to find a probability distribution over the possible models using the current parameters. And in the maximisation step (M step), new parameters are resolved using current probabilities. These new estimated parameters from the M step are then used to determine the probability distribution of the hidden variables in the following E step. After several iterations, the EM algorithm converges to a local maximum.

An equivalent of the EM algorithm is the Baum-Welch algorithm (Baum, et al., 1970) and it is one of the most commonly used re-estimation process for HMM training. However, other training algorithm such as the Segmental K-means algorithm (Juang & L.R., 1990) (Dugad & Desai, 1996) for estimating parameters in HMM is also very common. (Boyle & Hanlon, 1995)

2.1.1 Structure of HMM used:

Depending on the kind of movement and probable number of state of each activity can take; different HMM structures may be applied. To illustrate this, the walking movement can be represented using two states for instance by means ofan ergodic structure. One state denoting the left foot whiles the other one the right foot i.e. the states will transition back and forth as the person walks. However, for falling gestures the probability of transitioning to a previous state is less apparent. This kind of movement can have both an HMM with a left-right structure or an ergodic model. All computation was taken to have an HMM with ergodic structure.

2.2 Making the machine learn movements through training

The goal is to find the most likely HMM (number of states, likelihood and parameters) for a particular observation sequence. The decoding step, described before has been skipped in this work as we are concentrating in achieving the optimal number of states.

The human movements

One set of data consist of numerous (x, y) coordinates. This is measured by using the biaxial accelerometer, which is attached to the waist of the person. The four distinct type of human body movements considered in the project are:

1. Getting up when the person was already sitting and vice versa (sit-up) as in Figure 4.

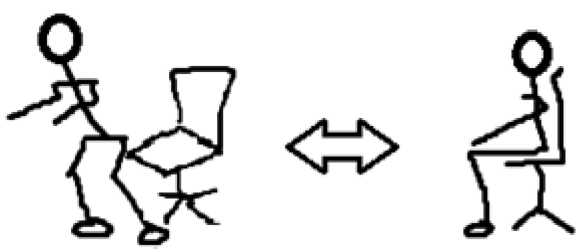

Figure 4 - Person doing action to sit down

2. A static fall while the person is standing and the latter collapses on the ground (Figure 5).

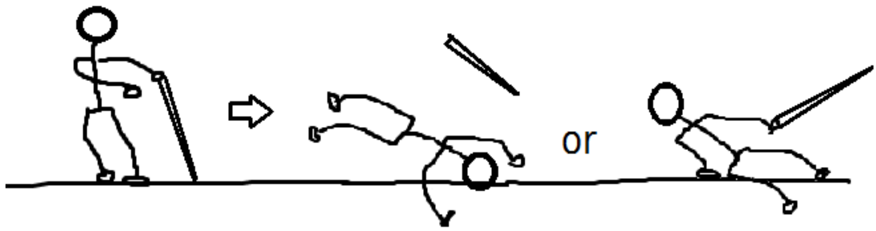

Figure 5 - Old person standing using a stick and falling either from the front or backwards

3. A dynamic fall while the person is walking, he/she falls down vigorously (Figure 6).

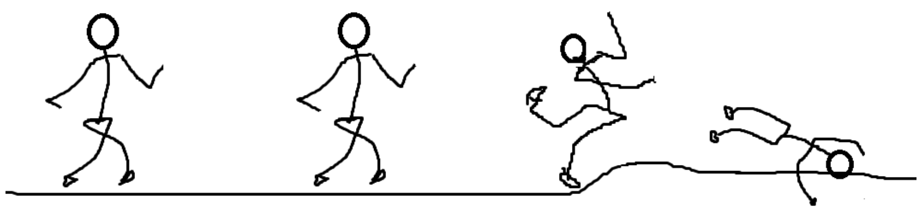

Figure 6 - Picture depicts the movement of a falling person while walking

4. .An individual is walking normally (**Figure 7**)

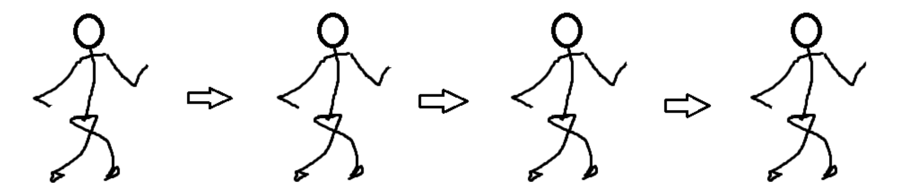

Figure 7: Person walking to the right

Two persons carried out the experiment (Al-Ani, et al., 2007). For each distinctive body gesticulation, the data is captured five times for both persons. These raw data collected by the accelerometer, were then further processed to extract relevant information contained therein. Two feature extraction technique, namely Laplacian smoothing and wavelets were employed directly on the raw data. Furthermore, data sequences that had already gone through Laplacian smoothing, were again processed through wavelet transformation. Therefore, for one person, four different training was carried out just for each movement. For each human motion, training was performed using:

- raw data sequences
- smoothed data sequences
- wavelet data sequences
- smoothed-wavelet data sequences

For every single body movement $(w = 1, \dots W)$ i.e. walking, fall while standing, etc. different sets of HMMs $\{\lambda_w\}$ are then built using the data collected by the accelerometer and from the feature extracted Laplacian smoothing and wavelet data sequences.

2.2.1 Laplace Smoothed data:

This makes use of an algorithm that tries to remove noise or other kind of fluctuations that corrupted the data points (Wikipedia, 2013). By doing so, the real pattern of the data can be extracted provided that the smoothing technique is performed reasonably (else the data pattern would be misjudged). Data smoothing can be achieved in many ways embracing random, random walk, moving average, simple exponential, linear exponential and seasonal exponential smoothing. Data smoothing can be interpreted as fitting a curve that best fits the data. The made-up graph on the left of Figure 8 is characterized by two variables. We can clearly observe that the data has many peaks and tough. This is a consequence of noise. Applying Laplacian smoothing, this noise can be removed by averaging neighboring data iteratively until finally getting an even curve as in the graph on the right in Figure 8.

Figure 8: Example of Laplacian Smoothing on a data sequence

2.2.2 Wavelets data:

Just like data smoothing, (Li, et al., 2010) data wavelet is a technique used to extract useful pattern of information from large amount of data. The raw data is always corrupted with noise, fluctuations or is incomplete. Wavelets are small waves whereby the term small means that the signal is broken into smaller parts and the waves means that these small parts are oscillatory waves. The Fourier Transform (FT) resembles wavelets in the sense that the signal is split into smaller units or segments terms of sine and cosine terms. However wavelet transforms (WT) are preferred because of their ability to represent the signal in the time and frequency domain simultaneously whereas FT represents the signal in the frequency domain only.

2.3 Recognition Stage

Now, for recognition phase, new HMMs are built with new observed data sequence. This new data is trained for the different sets of HMM, λ_w for all the four activities. For each hidden state in the test set, the probability $P_w = P(O|\lambda_w)$ of the observation sequence given the different models λ_w is calculated for each activity. To recognise an activity, we need to extract the model which maximises the likelihood $P(O|\lambda)$ pertaining to all HMMs. The unidentified observation, O can be defined as $w^* = arg \max_{1 \le w \le W} P_w$. Then the forward probability can be recursively used to determine the probability to be in state q_i at time t given the model λ_w. When comparing between two HMMs, there need to be a measure for the distance between them. One way to do that is by applying the Kullback-Leibler distance (Juang & Rabiner, 1985) which measures the difference between two probability distribution function to see how close they are (Dugad & Desai, 1996).

3. THE SEQUENTIAL PRUNING STRATEGY

(Bicego, et al., 2002) indicate that the EM algorithm is a computationally demanding process typically when the re-estimation starts using initial parameters. The algorithm needs to iteratively adjust new parameters until it converges to local maxima.

After having learned the parameters of the model, the Forward algorithm computes the likelihood of the model given the observed data. Again, to find the likelihood of other models (i.e. for different states), the learning procedure is once more started using the initial guess of the parameters.

The purpose of this paper is the improvement of the standard training process. Previously the training was carried out individually for different number of states and the one that gave the highest likelihood was the HMM which was the most representative for the observation sequence. For each state, the training started with a guess of initial parameters. An example for the initial transition matrix for a three state model and a two state model are shown below. The probability (ergodic structure) of going from one state to another is equally distributed before training. For any other HMM models with N states, both initial and transition probabilities are assigned equally everywhere and have a value of $1/_N$

$$\begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix} \qquad \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

3 states 2 states

For the emission matrix, the probabilities are calculated using the Gaussian distribution. Because the data from the sensor is made up of the horizontal movement (x-axis) and those of the vertical acceleration movement (y-axis), the probabilities are defined by through a bivariate Gaussian consisting of a mean vector matrix for each variable and a variance-covariance matrix having the variance of every variable and the covariance joining both variables.

The sequential pruning strategy makes the training's initialisation more sensitive when compared to traditional method of training an HMM. This new type of approach will train HMMs so as to obtain an accurate number of states and at the same time reduce the computational expense. The sequential pruning strategy is performed in a sequence of decreasing steps. The idea of decreasing learning strategy combined with the removal of the least probable state from each training session helps in finding more accurately the optimum state. Training is started using large number of states and consequently obtains the initial parameters. The same parameters will be used throughout the whole training algorithm.

The strategy which is a "decreasing" learning procedure whereby training starts from a large amount states (say 10 states) and terminates using a small number of states (1 state). Let us denote the first training to be that of the maximum number of states chosen by the user as K_{max} and the minimum number of states as K_{min}. When a large number of states are used, the estimated parameters would not be much affected by the initial parameters. Each data set is segmented or broken into the desired number of states using K-means algorithm (Moore, 2001). After segmentation, the initial value for the transitional probabilities is assigned for the model and output probabilities are calculated form the acquired data. At first, everything is assumed equally likely, like so the probability in the transition matrix is equal everywhere on every row and each row sums to one (stochastic matrix). After the training, new parameters for the model are obtained. These new parameters for K_{max} are then used for the next iteration as defined by the strategy.

The least probable state is pruned from the state transition matrix as well from the emission matrix. The state which needs to be pruned depends on the probabilities of the transition matrix since these values determine the structure of the HMM. Then, the least probable state from the other parameters (initial and emission matrices) of the HMM are taken out. Each training session begins using a more accurate estimate obtained by pruning the least probable state from the result of the previous training session. In this way a better accuracy is obtained for the optimal number of states whilst reducing the computational burden.

The computational complexity means that lesser number of iterations is entailed due to the effective initialisation process. The strategy stops until it has finished training the last model defined by the user i.e. K_{min}

3.1 Eliminating Least Probable State

Pruning the least probable state is done by removing rows and columns containing the most number of zeros from the transition matrix. The zero in the matrix signifies that there is no relationship amid two states. The transition matrix may have several, one or no zeros after the learning algorithm has trained the model. Examples of three (3×3) transition matrices having many zeros (Figure 9), one zero (Figure 10) or no zero (Figure 11) respectively.

$$\begin{pmatrix} 0.9523810 & 0.0178571 & 0.0297619 \\ 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 9: Numerous zeros in transition matrix

$$\begin{pmatrix} 0.9661889 & 0.0028915 & 0.0309197 \\ 0.0192758 & 0.9807242 & 0 \\ 0.0090048 & 0.0032584 & 0.9877368 \end{pmatrix}$$

Figure 10: One zero in transition matrix

$$\begin{pmatrix} 0.9032993 & 0.0485941 & 0.0481066 \\ 0.0126725 & 0.9345936 & 0.0527338 \\ 0.0038539 & 0.0187844 & 0.9773617 \end{pmatrix}$$

Figure 11: No zeros in transition matrix

For the case of numerous zeros, the row where there is maximum amount of zeros is removed. Subsequently, the corresponding column is also removed. Looking at Figure 9 the third row has two zeros compared to the others. This row is eliminated and at the same time the third column is taken out. Note that in the case a column has got more zeros than a row, the column is erased first and then the corresponding row.

For the case of Figure 10, the matrix has one zero. Consequently, the row where the zero is found is removed and so is the matching column. That is the second row and column is pruned. However, when no zeros are present in a matrix (Figure 11) the minimum probability is spotted and in this case the third row and column is eliminated.

As for the mean matrix, the removal of its column depends on what row is taken out from its corresponding transition matrix. The same procedure is made for the covariance matrices. For a three state model, there will be three covariance matrices each of which is a (2×2) matrix. Taking as example, the transition matrix of Figure 9 again, the third covariance matrix will be removed.

3.2 Model Selection Criterion

Maximum likelihood criterion generally over-estimates the model given the observations. Thus, there need to have a penalty term to remedy this problem in model selection. (Roblès, et al., 2011). A number of models selection criterions have been presented in the statistics of literature. Bayesian Inference Criterion (BIC) (Schwarz, 1978), Minimum Description Length (MDL) (Rissanen, 1986), Akaike's Information Criterion (AIC) (Akaike, 1974) or Bayesian Ying-Yang harmony (BYY) (Hu & Xu, 2004) are models selection criteria. BIC has been proposed as a model selection criterion in this work. It consists of a combination of two terms: a maximised likelihood (ML) term and a penalty term, which in turn depends on the number of free parameters (i.e. total number of zeros in the transition matrix) of the model and the number of data sequences recorded for each activity. The likelihood tends to increase as the model size increases and so does the penalty term.

Thus, the penalty term penalises larger models and helps to find the optimal model from a set of available models. The optimum number of states is the maximiser $k_{BIC} = argmax\ BIC\ (k)$ where

$$BIC\ (k) = \log P(O|\lambda_k) - {}^{N_k}/_{2} \log(n) \quad \textbf{(4)}$$

In Equation 4, $\log P(O|\lambda_k)$ denotes the log-likelihood estimate of the model with k states, N_k is the total number of free parameters and n is the number of observations.

N_k is determined as follows (Sergey, 2011) :

$$N_k = [No.\ of\ states\ \times (No.\ of\ states - 1)] - Total\ number\ of\ zeros\ in\ transition\ matrix \quad \textbf{(5)}$$

For every iteration in the sequential pruning strategy, the BIC criterion is computed using the current parameters and the value is stored in an array. After completion of the algorithm, the highest value of the BIC criterion for whatever model it corresponds is taken to be the optimal state.

The Bayesian Ying-Yang harmony (BYY) has better performance for model selection in case of small observed data sets. This algorithm performs rather well for small data sets compared to other model selection criteria such as AIC, MDL, and BIC.

All the computations were carried out in an HMM toolbox (Al-Ani & Hamam, 1996) in a Scilab® environment. The pruning strategy is summarised in the flow chart of Figure 12.

3.3 The State Splitting strategy

The same experiment was repeated (Beeharry, 2011) using the similar data but adopting the State Splitting Strategy (Stenger, et al., 2001) coupled with the

Minimum description Length (MDL) (Rissanen, 1986) as model selection criterion.

Training is started from state 1 and if the likelihood of this HMM model does not satisfy the MDL criterion, the model is split into 2 states etc., to find the best model. However, this strategy does not deal with the problem of initialisation and computational burden. MDL criterion is shown in Equation 5.

$$MDL = Log\ L(\lambda_{k+1}|O) - Log\ L(\lambda_k|O) > (N + \frac{d^2+3d}{4}) \qquad \textbf{(6)}$$

Where: $L(\lambda_{k+1}|O)$ is the likelihood of present state; $Log\ L(\lambda_k|O)$ is the likelihood of previous state; $(N + \frac{d^2+3d}{4})$ is the penalty part and N depends on the dimensionality of the HMM.

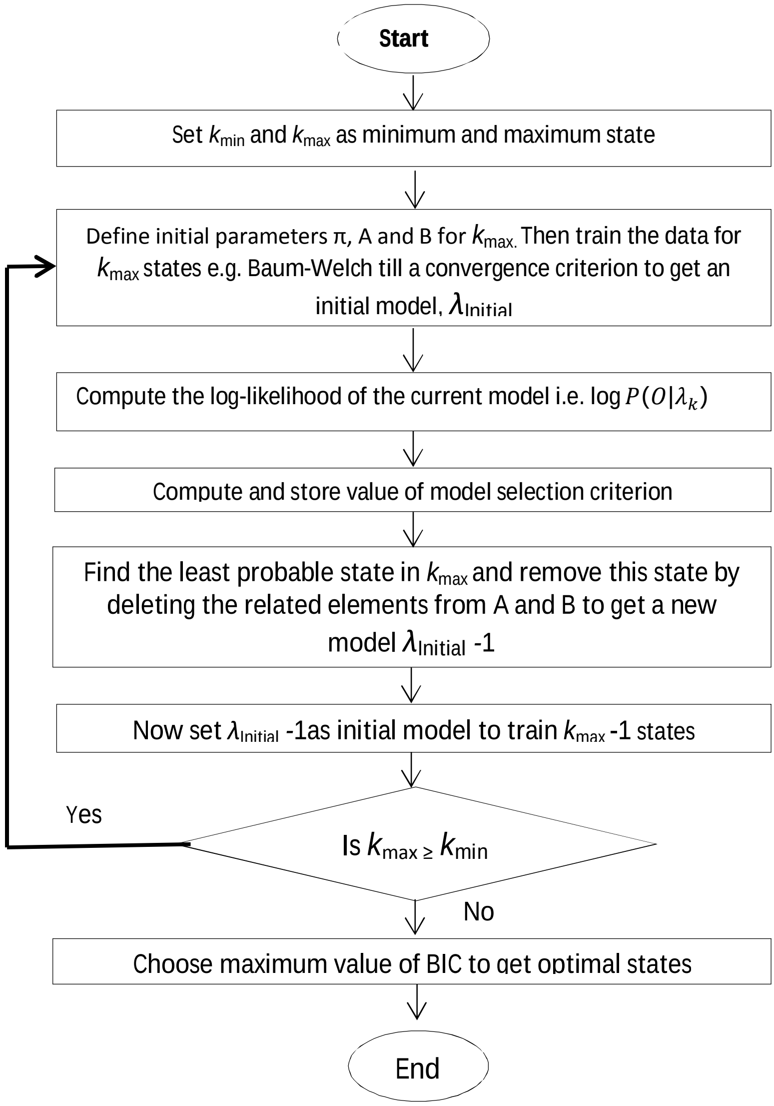

Figure 12: Flow chart showing decreasing learning strategy through sequential pruning of HMM parameters

The Sit/Up event is used in order to demonstrate how the proposed algorithm works in reality. The smoothed Laplace data of the raw signal is taken here.

0.944148000000000	-0.328682000000000
1.11930971878098	-0.329507573293341
1.15534483082837	-0.331788266070647
1.08240431676521	-0.337023754344817
1.01064769596514	-0.346000381949375
0.979089392720459	-0.358859518746570
0.965111632248596	-0.371971431209065
0.953827179683696	-0.372720842876892
0.946521238670269	-0.364210275894680
0.939707587811925	-0.367987935473890
0.935603001224516	-0.394644267441748
0.926139452597547	-0.426088876663571
0.902322448955082	-0.429159857061063
0.881550676479185	-0.425131142483781
0.886162050147295	-0.446162097874657
0.891331782290973	-0.464102456314829

Figure 13 – Part of Raw data acquired by accelerometer while person is performing sit/up movement. This data has been smoothed by Laplace.

The sequence of data is fed into the Scilab toolbox and is trained using the Sequential decreasing strategy. Training is started with five states. This resulted as tabulated below:

Table 1 - Training data with 5 states

Parameter	value
Initial state probability, π	[1 0 0 0 0]
Transition matrix, A_{ij}	[0.927,1.48× 10^{-2},5.79× 10^{-2},0.00,0.00; 6.35× 10^{-2},0.902,2.65× 10^{-2},0.00,8.04× 10^{-3}; 3.61× 10^{-2},2.10× 10^{-2},0.922,0.00,2.09× 10^{-2}; 0.00,0.00,0.00,0.919,8.15× 10^{-2}; 0.00,0.00,3.17× 10^{-2},1.44× 10^{-2},0.954]
Mean vector, Me	[0.904,0.965,1.01,0.910,0.924; -0.336,-0.432,-0.233,0.147,-1.06× 10^{-2}]

	$[2.85\times 10^{-3}, 1.24\times 10^{-3};$ $1.24\times 10^{-3}, 2.78\times 10^{-3}]$,.. $[2.81\times 10^{-3}, 6.08\times 10^{-4};$ $6.08\times 10^{-4}, 4.57\times 10^{-3}]$,.. $[1.15\times 10^{-2}, -6.15\times 10^{-3};$ $-6.15\times 10^{-3}, 6.52\times 10^{-3}]$,.. $[3.07\times 10^{-4}, 1.89\times 10^{-4};$ $1.89\times 10^{-4}, 3.17\times 10^{-3}]$,.. $[1.37\times 10^{-2}, -3.30\times 10^{-3}$ $-3.30\times 10^{-3}, 3.75\times 10^{-3}]$
Maximum likelihood estimate, LP	11.6

Table 2 - Training with 4 states after pruning least probable state from the 5 state parameters

Parameter	value
Initial state probability, π	[1 0 0 0]
Transition matrix, A_{ij}	$[0.925, 1.51\times 10^{-2}, 6.04\times 10^{-2}, 0.00;$ $7.20\times 10^{-2}, 0.901, 1.88\times 10^{-2}, 8.42\times 10^{-3};$ $3.75\times 10^{-2}, 1.95\times 10^{-2}, 0.923, 1.97\times 10^{-2};$ $0.00, 0.00, 2.57\times 10^{-2}, 0.974]$
Mean vector, Me	$[0.908, 0.956, 1.01, 0.921;$ $-0.334, -0.443, -0.233, 1.28\times 10^{-2}]$
Covariance matrices, Seg	$[3.00\times 10^{-3}, 1.09\times 10^{-3};$ $1.09\times 10^{-3}, 2.50\times 10^{-3}]$,.. $[3.16\times 10^{-3}, 5.51\times 10^{-4};$ $5.51\times 10^{-4}, 4.07\times 10^{-3}]$,.. $[1.15\times 10^{-2}, -6.14\times 10^{-3};$ $-6.14\times 10^{-3}, 6.58\times 10^{-3}]$,.. $[1.20\times 10^{-2}, -3.26\times 10^{-3};$ $-3.26\times 10^{-3}, 7.48\times 10^{-3}]$
Maximum likelihood estimate, LP	11.0

Table 3 - Training with 3 states after pruning least probable state from the 4 state parameters

Parameter	value
Initial state probability, π	[1 0 0]
Transition matrix, A_{ij}	[0.920,3.89$\times 10^{-2}$,4.12$\times 10^{-2}$; 6.35$\times 10^{-2}$,0.911,2.57$\times 10^{-2}$; 2.52$\times 10^{-2}$,7.21$\times 10^{-3}$,0.968]
Mean vector, Me	[0.920,0.958,0.964; -0.304,-0.415,-9.33$\times 10^{-2}$]
Covariance matrices, Seg	[2.23$\times 10^{-3}$,4.95$\times 10^{-4}$; 4.95$\times 10^{-4}$, 2.47$\times 10^{-3}$],.. [7.61$\times 10^{-3}$,2.59$\times 10^{-3}$; 2.59$\times 10^{-3}$, 4.57$\times 10^{-3}$],.. [1.46$\times 10^{-2}$,-1.05$\times 10^{-2}$; -1.05$\times 10^{-2}$,2.08$\times 10^{-2}$]
Maximum likelihood estimate, LP	9.49

Table 4 - Training with 2 states after pruning least probable state from the 3 state parameters

Parameter	**value**
Initial state probability, π	[1 0]
Transition matrix, A_{ij}	[0.959,4.06$\times 10^{-2}$; 3.45$\times 10^{-2}$,0.966]
Mean vector, Me	[0.927,0.968; -0.344,-0.101]
Covariance matrices, Seg	[3.89$\times 10^{-3}$,4.28$\times 10^{-4}$; 4.28$\times 10^{-4}$, 6.05$\times 10^{-3}$],.. [1.49$\times 10^{-2}$,-1.13$\times 10^{-2}$; -1.13$\times 10^{-2}$,2.22$\times 10^{-2}$]
Maximum likelihood estimate, LP	8.73

Table 5 - Training with 1 state after pruning least probable state from the 2 state parameters

Parameter	**value**
Initial state probability, π	[1]
Transition matrix, A_{ij}	[1]
Mean vector, Me	[0.949;-0.213]
Covariance matrices, Seg	[1.05×10^{-2},-3.60×10^{-3}; -3.60×10^{-3},2.96×10^{-2}]
Maximum likelihood estimate, LP	6.02

Using equation 4, the BIC values for 5 consecutive states are: 7.73 (5 states); 7.89 (4 states); 7.39 (3 states); 8.20 (2 states); 6.02 (1 state)
Optimum state chosen: 2

4. RESULTS

The purpose of the experiment is to make the machine learn the difference amongst the four activities and to distinguish between each of them. The events: walk, fall and sitting down- getting up were performed by the two persons. The measured data is collected and run into the HMM toolbox in Scilab software. Before that, the raw data is processed to remove noise or fluctuations. Three different approaches including Laplace smoothing, wavelet transform and smoothed wavelet transform are applied on the measured data. Wavelet transform is a technique used to extract useful pattern of information.

Then, training is performed for all data types including raw data. The training is done using the Baum-Welch algorithm and the Viterbi based (Segmental K-means) algorithm which is inbuilt from the HMM toolbox. On average the HMMs built for the two persons for the same events are: 2 states for walking and sitting down-getting up events, 4 states for fall while walking event and 5 states for fall while standing event. The graph of Figure 14 shows the optimal states for each event using the different types of data for one person using Baum-Welch algorithm.

Figure 14: Training using Baum-Welch algorithm for one person using the different data

All other results follow the same general trend as seen from Figure 14 with sitting and walking movements requiring fewer states than that of the falling movements. It was also observed that both falling events were represented by more complex models. These complex models had much more free parameters in their transition matrix than the others. Even with the penalty of BIC, the high likelihood of complex states generated 4 to 5 optimum states. Moreover, the optimum number of states for the two persons differed more or less by one state. This can be explained by the fact that while performing the experiment the two persons' movement for a particular event cannot be exactly same.

It was also seen that Segmental k-means algorithm outputted erroneous results when training Laplacian smoothed data and wavelet data. This was due the small

number of data points for each event. This is obvious because data smoothed and data wavelet removed noise and random fluctuations from the raw data.

Compared to the original work of (Al-Ani, et al., 2007), both walking and sitting events needed 2 states whereas fall while standing and fall while walking were modelled using 2 and 3 states respectively. As for the State Splinting strategy with MDL as model selection criterion, the fallouts were same as that that of (Al-Ani, et al., 2007). The HMM with 2 states for walking and sitting have a structure can be regarded as ergodic since the present state returns to the previous state (e.g. sit down-get up, or left foot, right foot, left foot, etc.) i.e. any state can be reached from whatever state with a single transition in the transition matrix.

There is a starting state, a transient state and a final state. In this work, the structure for all HMMs is ergodic. In contrast to Al-Ani and his associates' work which made use of the standard training method, the same number of states was obtained for walking and sitting events. However, both falling events, the HMMs consisted of 2 to 3 states. With the sequential pruning strategy, the falling events are made up of more states. We can say that the new method give a more precise HMM as the same event can be characterised using higher number of states.

5. CONCLUSION

Fall is a leading cause of sufferance among old people. There is a need to provide for an automatic on-line detection of people fallen down especially those who live alone at home. With this new technique for improving the training of HMMs, detection of events such as walk, fall and sitting down- getting up should be much faster and more accurate. The results obtained are especially encouraging. For the time being, all these computations are being simulated through a PC with the data coming via a wireless link from the sensor. The new strategy in this work showed how to minimise the complexity of the training procedure and at the same time producing positive results. In the near future, we hope to all these calculation done in our smart phones or other miniature devices that will detect and alert emergency services to save lives.

5.1 Future Works

- Much focus in this work was on finding the optimum model for the four kinds of activities using a new training method. This can be now put into practice for detecting any of the four human movements in real time.

- Multiply the training procedure with more test data for each of the four different movements.

- Simulate the different movements by more than two people. For example, the falling gesture will never be the same for every person. Training using more data will allow for better recognition later on.

- Use of different model selection criterion for training such as MDL or BYY and compare the results to see how they differ.

- (Churbanov & Winters-Hilt, 2008) proposes a more efficient training with the Baum-Welch algorithm which takes less processing time and (Romeijn, et al., 2012) the new BIC criterion called the prior-adapted BIC which selects the model according to the likelihood of the model given the data, the complexity and model size. Whereas the normal BIC, selects the model according only to the likelihood of the model given the data, and the complexity. Merging the propositions of these two papers along the sequential pruning strategy can help to find the optimal state.

6. REFERENCES

Akaike, H., 1974. A new look at the statistical model identification. IEEE *Transactions on Automatic Control,* December, 19(6), pp. 716-723.

Al-Ani, T. & Hamam, Y., 1996. *An Integrated Environment for Hidden Markov Models – A Scilab Toolbox.* Dearborn, MI, Computer-Aided Control System Design, 1996., Proceedings of the 1996 IEEE International Symposium on, pp. 446 - 451.

Al-Ani, T., Monacelli, E. & Le Ba, Q. T., 2007. *On-line Automatic Detection of Human Activity in Home using Wavelet and Hidden Markov Models Scilab Toolkits.* Singapore, IEEE, pp. 485 - 490.

Analog Devices, n.d. *ADXL202E Low Cost ±2gDual-Axis Accelerometer with Duty Cycle Output,* s.l.: Data Sheet.

Baum, L. E., 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities,* Volume 3, pp. 1-8.

Baum, L. E., Petrie, T., Soules, G. & Weiss, N., 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics,* February, 41(1), pp. 164-171.

Beeharry, Y., 2011. *State slpitting techniques for Hidden Markov Model (HMM) selection and its application to the modeling of human movement,* Réduit, Mauritius: University of Mauritius.

Bicego, M., Murino, V. & Figueiredo, M. A., 2002. A sequential pruning strategy for the selection of the number of states in Hidden Markov models. *Pattern Recognition Letters,* 22 Feb, 24(9), pp. 1395-1407.

Boyle, R. D. & Hanlon, S., 1995. *Hidden Markov Models - an online tutorial.* [Online]
Available at:
http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html
[Accessed 05 March 2013].

Burge, C. & Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology,* Volume 268, pp. 78-94.

Centers for Disease Control and Prevention, 2012. *Falls Among Older Adults: An Overview.* [Online]
Available at:
http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html
[Accessed 10 July 2013].

Churbanov, A. & Winters-Hilt, S., 2008. Implementing EM and Viterbi algorithms for Hidden Markov Model in linear memory. *BMC Bioinformatics,* 30 April.Volume 9:224.

Colorado State University, 2012. *Preventing Falls in the Elderly.* [Online] Available at: Internet:
http://www.ext.colostate.edu/pubs/consumer/10242.html
[Accessed 27 November 2012].

Dempster, A., Laird, N. & Rubin, D., 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological),* 39(1), pp. 1-38.

Dugad, R. & Desai, U., 1996. *A Tutorial on Hidden Markov Models,* Mumbai: Indian Institute of Technology.

Forney, G. . D. J., 1973. The Viterbi algorithm. *Proceedings of the IEEE,* March, 61(3), pp. 268-278.

Hu, J., Brown, M. K. & Turin, W., 1996. HMM based online handwriting recognition. *IEEE Transactions, Pattern Analysis and Machine Intelligence,* 18(10), pp. 1039-1045

Hu, X. & Xu, L., 2004. Investigation on several Model Selection Criteria for determining the Number of cluster,". *Neural Information Processing-Letters and Reviews,,* 4(1), p. July

Juang , B.-H. & L.R., R., 1990. The segmental K-means algorithm for estimating parameters of hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions,* September, 38(9), pp. 1639 - 1641.

Juang, B.-H. & Rabiner, L. R., 1985. A Probabilistic Distance measure berween HMMs. *AT&T Technical Journal,* February, 64(2), pp. 391-408.

Learn Not To Fall, 2012. *How Often Falls Occur.* [Online] Available at: http://www.learnnottofall.com/content/fall-facts/how-often.jsp
[Accessed 10 July 2013].

Li, T., Ma, S. & Ogihara, M., 2010. Wavelets Methods in Data Mining. In: O. Maimon & L. Rokach , eds. *Data Mining and Knowledge Discovery Handbook.* s.l.:Springer US, pp. 553-571.

M.Tineti, D. I. Bake, G. McAvey & E.B. Claus, P. G, 1994. *A Multifactorial Intervention to Reduce the Risk of Falling among Elderly People Living in the Community.* [Online]

Available at: http://www.nejm.org/doi/full/10.1056/NEJM199409293311301#t=articleMethods
[Accessed 27 November 2012].

Microchip, n.d. *PIC16F87XA,* s.l.: Microchip Technology Inc..

Moore, A. W., 2001. *K-means and Hierarchical Clustering.* [Online]
Available at: http://www.autonlab.org/tutorials/kmeans11.pdf
[Accessed 01 December 2012].

Peshkin, L. & Gelfand, M. S., 1999. Segmentation of Yeast DNA Using Hidden Markov Models. *Bioinformatics ,* 15(12), pp. 980-986.

R.Rabiner, L., 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings IEEE,* 77(2), pp. 257-286.

Rissanen, J., 1986. Stochastic complexity and modelling. *Annals of Statistics,* 14(3), pp. 1080-1100.

Roblès, B. et al., 2011. *Evaluation of relevance of stochastic parameters on Hidden Markov Models.* Troyes, France, s.n.

Romeijn, J.-W., Schoot, . R. v. d. & Hoijtink, H., 2012. One Size Does Not Fit All: Proposal for a Prior-adapted BIC. In: W. J. G. S. H. M. S. M. W. Dennis Dieks, ed. *Probabilities, Laws, and Structures.* s.l.:Springer Netherlands, pp. pp 87-106.

Schwarz, G., 1978. Estimating the dimension of a model. *The Annals of Statistics,* March, 6(2), pp. 461-464.

Sergey, 2011. *Number of parameters in Markov model.* [Online]

Available at: http://stats.stackexchange.com/questions/12341/number-of-parameters-in-markov-model/44328#44328
[Accessed 25 August 2012].

Ślusarczyk, G. & Augustyniak, P., 2010. A Graph Representation of Subject's Time-State Space. In: E. Piętka & J. Kawa , eds. *Information Technologies in Biomedicine.* s.l.:Springer Berlin Heidelberg, pp. 379-390.

Stenger, B. et al., 2001. *Topology Free Hidden Markov Models: Application to Background Modeling.* Vancouver, IEEE, pp. 294-301.

Wikipedia, t. f. e., 2013. *Laplacian smoothing.* [Online]
Available at: http://en.wikipedia.org/wiki/Laplacian_smoothing
[Accessed 10 October 2013].

Zheng, X. & Koenig, S., n.d. *A Project on Gesture Recognition with Neural Networks for "Introduction to Artificial Intelligence" Classes.* [Online]
Available at: http://idm-lab.org/bib/abstracts/papers/project3.pdf
[Accessed 11 July 2013].