North Carolina State University, ECE Department

Project 2 Report

Zexi Chen, Qian Ge, Hui Guan

December 5, 2015

1 Introduction

The task of activity recognition from wearable sensing has received increasing attention in recent years, resulting from the latest sensor technology development and application demands. Activity recognition plays an important role in tracking human behavior , supporting health care monitoring and performing social interaction. This project focuses on high-level activity recognition from wearable sensing. A benchmark dataset from the OPPORTUNITY Activity Recognition Challenge [2] is used to validate our proposed approaches. High-level activities we aim to predict involve *Relaxing*, *Coffee time*, *Early morning*, *Cleanup*, *Sandwich time* and a *NULL* class. Only 12 bluetooth 3-axis accelaration sensors are taken into account to form an input space dimensionality of 36 for our task.

Sensor-based activity recognition is a classification problem where both generative methods that modeling the underlying distribution of each activity and determinative techniques that learning decision boundaries have been applied. [3] used boosting methods for feature selection and then applied ensemble classifications to generate inputs into HMM models. The hybrid approach showed that applying HMM on-top of discriminative classifiers can not only effectively improve the performance but also smooth outputs. [2] tested four classification techniques including K-Nearst Neighbours(KNN), Nearest Centroid Classifer, Linear Discriminant Analysis and Quadratic Discriminant Analysis to provide a baseline performance, and found that KNN performs the best for both locomotion and gestures recognition.

In this project, to obtain the discrete observation data, we first cluster the continuous raw sensor data into several groups by Gaussian Mixture Model (GMM). Then we use two approaches to do the high-level activity recognition: (1) Model high-level activity transition using one Hidden Markov Model (HMM). The activity sequence is then estimated by the Viterbi path of the HMM. (2) Model each high-level activity by one HMM. Each HMM is trained by the observation of a specific high-level activity. When testing, we divided the test sequence into several fixed length segments and the activity of each segment is estimated by the HMM with maximum likelihood.

2 METHODOLOGY

2.1 OVERVIEW

Our methodology consists of two parts: the first part is the feature extraction and transformation work to generate inputs suitable for human activity modeling and the second one is the application of HMM to capture temporal dynamics of human activities. We proposed two frameworks which are identical in feature processing but different in the way activities are modeled. The frameworks are shown in Figure 2.1 and 2.2 and will be further explained.

Figure 2.1: Framework for a single HMM model

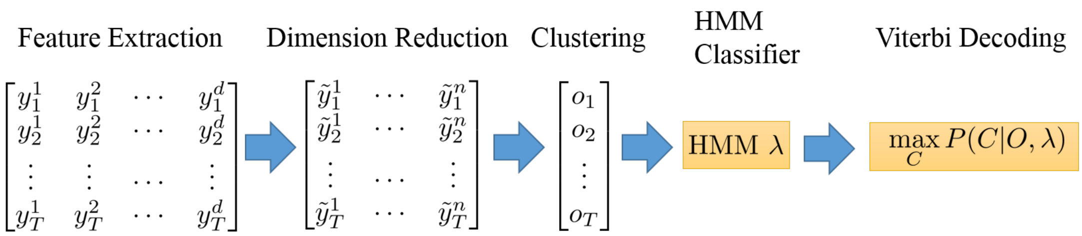
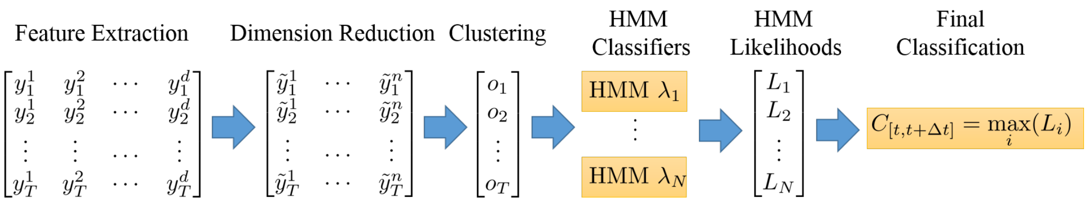


Figure 2.2: Framework for multiple HMM models



2.2 FEATURE PROCESSING

Preprocessing steps contain feature extraction, dimension reduction and GMM clustering. Raw data collected directly from sensors usually contain a considerable amount of missing value. We due with the missing value in two ways:

1. If the number of missing values takes up more than one third of the total number of sensor values in a specific time, the data sample is removed.

2. Else, repeat the previous value in place of the missing value if previous value exists.

Since the data was collected continuously, we used a sliding window of length 500 ms with a step size of 250 ms for extracting the features.Then we used as a feature the mean value of the sensor on each window, denoted as $\mathbf{y}_t = [y_t^1, y_t^2, \cdots, y_t^d]$. After feature extraction, we applied Principle Component Analysis to reduce our feature space to n dimensions, denoted as $\tilde{\mathbf{y}}_t = [\tilde{y}_t^1, \tilde{y}_t^2, \cdots, \tilde{y}_t^n]$. We chose the principle components whose eigenvectors explain 90% of total variance.

To discrete our features, We took advantage of GMM to form a total of M clusters, each corresponding to a kind of observation. Let $V = \{v_1, v_2, \cdots, v_M\}$ be M observation. Thus, the final feature sequence is $O = o_1 o_2 \cdots o_T, o_t \in V$. The above three steps are the same for both frameworks shown in Figure 2.1 and 2.2.

2.3 Model Training

HMMs have been successfully applied in modeling sequential data such as time-series in speech recognition and gesture tracking. We used HMMs to capture the temporal dynamics by taking as input labels sequences from our GMM clustering. We firstly describe the notations of HMM model and the model training process, and then explain in detail the difference of our single HMM model and multiple HMM models.

Let $S = \{S_1, S_2, \cdots, S_N\}$ be N hidden states and $A = \{a_{ij}\}, 0 \le i, j \le N$, be the state transition probability distribution, where each $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ indicating the probability of transition from state S_i at time t to state S_i at time $t+1$. Let $V = \{v_1, v_2, \cdots, v_M\}$ be M observation for each state. The observation probability distribution $B = b_j(k), 0 \le j \le N, 1 \le k \le M$, where each $b_j(k) = P(v_k | q_t = S_j)$ indicating the probability of observing v_k at time t given the state S_j. Notice that both the transition matrix A and the emission matrix B has the constraint $\sum_{j=1}^{N} a_i j = 1$ and $\sum_{k=1}^{N} b_j(k) = 1$. The initial state probability distribution is denoted as $\pi = \{\pi_i\}, 1 \le i \le N$, where $\pi_i = P(q_1 = S_i)$ and $\sum_1^n \pi_i = 1$. Then a HMM model is a triplet $\lambda = (A, B, \pi)$.

Given an observation sequence $O = o_1 o_2 \cdots o_T, o_t \in V$, the model training problem is to find an optimum model parameter vector $\lambda \in \Lambda$ that maximize $P(O|\lambda)$, $\lambda^* = \arg\max_\lambda P(O|\lambda)$. Our model is trained using Baum-Welch algorithm [1], an EM algorithm based on forward-backward procedure.

1. **Forward Procedure**
   Let $\alpha_t(i) = P(o_1, o_2 \cdots o_t, q_t = S_i | \lambda)$, the probability of observation sequence $o_1, o_2 \cdots o_t$ with state S_i given model λ. This is solved recursively:
   - **Initialization:** $\alpha_1(i) = \pi_i b_i(o_1), 1 \le i \le N$,
   - **Induction:** $\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right) b_j(o_{t+1}), 1 \le t \le T$.
2. **Backward Procedure**
   Let $\beta_t(i) = P(o_{t+1}, o_{t+1} \cdots o_T | q_t = S_i, \lambda)$, be the probability of observation sequence o_{t+1},o_{t+1} $\cdots$ o_T with state S_i given model λ. This is also solved recursively:
   - **Initialization:** $\beta_T(i) = 1, 1 \le i \le N$,
   - **Induction:** $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), 1 \le t \le T-1, 1 \le i \le N$.
3. **Update**
   The probability of being in state S_i at time t and state S_j at time $t+1$ given the observation sequence O and the model λ is:

$$\begin{aligned}\xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)},\end{aligned}$$

   The probability of being in state S_i at time t given the observation sequence O and the model λ is:

$$\begin{aligned}\gamma_t(i) &= P(q_t = S_i | O, \lambda) \\ &= \sum_{j=1}^{N} \xi_t(i,j).\end{aligned}$$

Then the parameters a_{ij}, $b_j(k)$ and π_i are solved by the equations:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{\gamma_t(i)}}, 1 \le i \le N, 1 \le j \le N,$$
$$\bar{b}_j(k) = \frac{\sum_{t=1,o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, 1 \le j \le N, 1 \le k \le M,$$
$$\bar{\pi}_i = \gamma_1(i), 1 \le i \le N.$$

For the framework shown in Figure 2.1, a single HMM model λ is trained to model human activity sequence with each activity being a state, namely, S_i is the i-th activity type. The number of hidden states N is the number of types of high-level activities and the observations are the training sequence $O = o_1 o_2 \cdots o_T, o_t \in V$. During testing, viterbi decoding decides the state assignment for each observation sample. Due to the limit of space, we didn't explain the algorithm here. Please refer to [4] for details of the algorithm.

In our multiple HMM models shown in Figure 2.2, each activity type is modeled with an individual HMM model. Given the number of activity types be N, the HMM models we trained are $\lambda_1, \lambda_2, \cdots, \lambda_N$. During testing, we have a continuous sequence S, which we use to compute the likelihood value $L_i([t, t+\Delta t])$ for λ_i at the time t using a sliding window of duration Δt:

$$L_i([t, t+\Delta t]) = P(O_{[t,t+\Delta t]} | \lambda_i, S)$$

The final segmentation and classification is based on the HMM that has the highest likelihood value,namely, $C_{[t,t+\Delta t]} = \max_i(L_i)$.

3 Results

Feature extraction, HMM training and decoding are implemented in MATLAB. For each person, we use ADL1-ADL3 as training set and ADL4-ADL5 as testing set. The high-level activities are relaxing, coffee time, early morning, cleanup and sandwich time (label 1-5) including a Null-class (label 6). The number of cluster used for discretization, $M = 50$, is manually chosen for both methods. For multiple HMM models, the number of hidden state for each HMM is chosen by cross-validation. The length of test segment, Δt, is 100.

We evaluate our result use the weighted F-measure, which is defined by

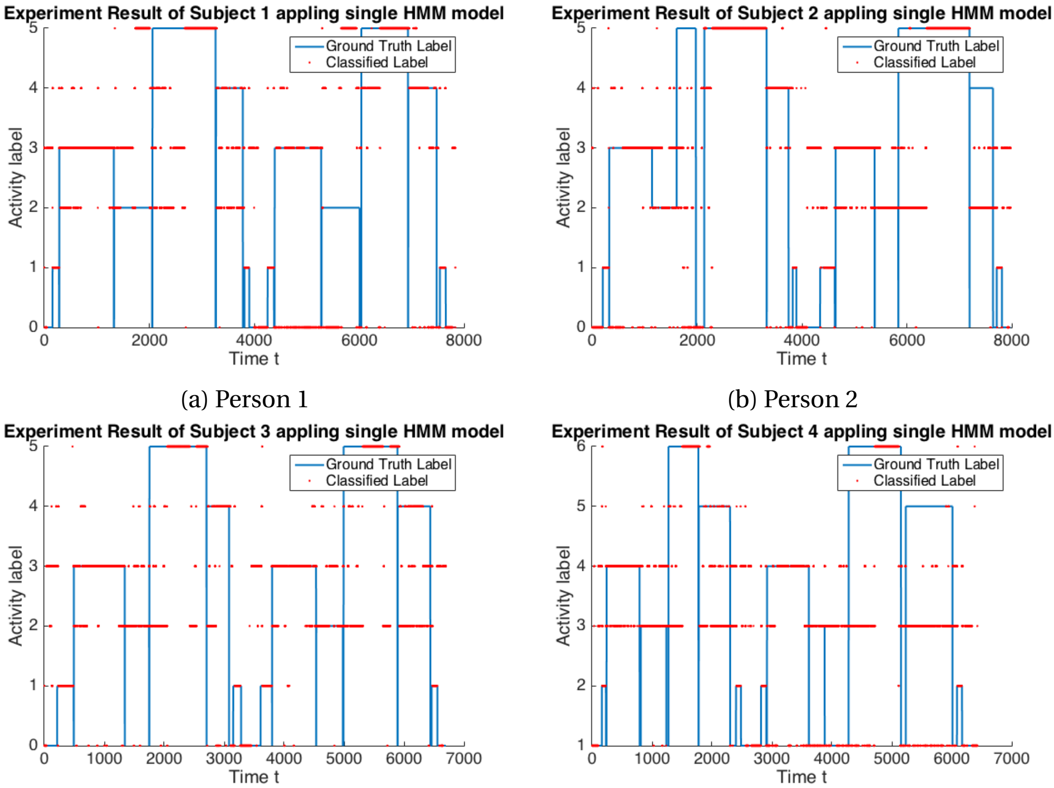
$$F = \frac{1}{N} \sum_{i}^{N} w_i F_i$$
$$F_i = 2 \frac{precision_i \times recall_i}{precision_i + recall_i}$$
$$precision_i = \frac{tp_i}{tp_i + fp_i}$$
$$recall_i = \frac{tp_i}{tp_i + fn_i}$$

where $N = 6$ is the number of high-level activities, w_i is the proportion of samples of class i, tp_i is true positive, fp_i is false positive and fn_i is false negative for each activity i.

For multiple HMM models, the number of hidden states N_i, $i = 1, \ldots, 6$ can be different for different activity. This is chosen by cross-validation. The training data is divided into three parts, ADL1-ADL3 and do 3-fold cross-validation. For each person, we first set $N_i = 2$, $\forall i$. Then change N_1 from 2 to

Table 3.1: F-measure of single HMM and multiple HMM

	S1	S2	S3	S4
Single HMM	0.4788	0.5699	0.5644	0.4106
Multiple HMM	0.6361	0.8119	0.8200	0.6498



(a) Person 1

(b) Person 2

(c) Person 3

(d) Person 4

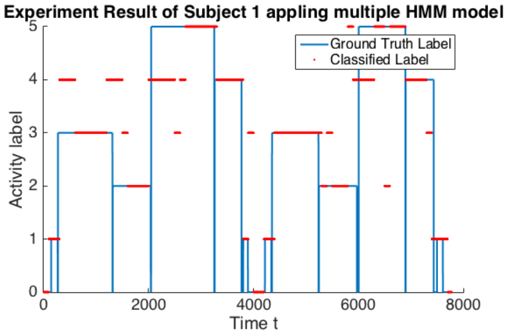
Figure 3.1: Classification result of single HMM model. Blue line is groundtruth and red dot is estimation result.

8 and pick the value with the highest F-measure as the best number of hidden state for activity 1. Repeat this process for $i = 2, \ldots, 6$ and the optimal N_i, $i = 1, \ldots, 6$ is obtained.

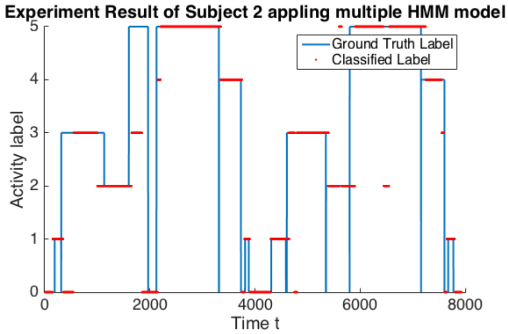
We test the activity classification using the two methods mentioned above. Fig. 3.1 and Fig. 3.2 show the classification results using single HMM model and multiple HMM models, respectively. Table 3.1 shows the weighted F-measure of each person for both methods. From the result, the performance of multiple HMM models is much better than single HMM model. That is, the multiple HMM models is more suitable for this application. This is because the transition of high level activity is not predictable. For example, after coffee time, the person can do any other activities. However, the state transition within a high level activity is predictable because each high level activity has a relatively stable pattern. The weighted F-measure of person 1 is low because the last three column is NaN. The low score of person 4 is because of the noisy data of that person.

4 CONCLUSION

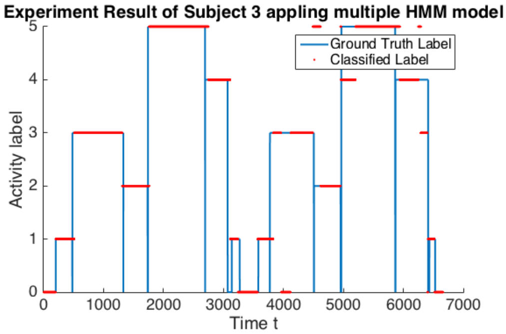
This paper has presented one single HMM model and one multiple HMM model to recognize human activities. Experiments on 4 subjects' daily activities show that the multiple HMM model performs much better than the single HMM model. When preprocessing the data, the process of missing data



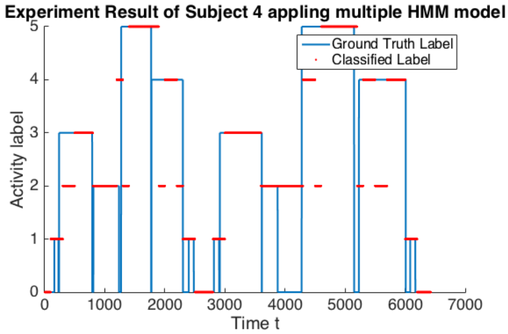
(a) Person 1



(b) Person 2



(c) Person 3



(d) Person 4

Figure 3.2: Classification result of multiple HMM models. Blue line is groundtruth and red dot is estimation result.

and feature selections play important roles on the prediction results. Our intention for further work will focus on missing data analysis and feature selection. Also, we will extend the HMM model from one-level to multiple-levels to incoporate more activity actions.

REFERENCES

[1] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, pages 164–171, 1970.

[2] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

[3] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, volume 5, pages 766–772. Citeseer, 2005.

[4] Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.