

PROJECT2. Human Action Recognition using Hidden Markov Model

Sangjun Son
Seoul National University
Department of Computer Science and Engineering
lucetre@snu.ac.kr

1. Introduction

Hidden Markov Model has such a good strength in analysing sequential data, and had been widely used in language model, part-of-speech tagging and named entity recognition [1]. Based on Markov process, HMM infers probability distributions of the model from observations and find the most likely results so far. It is useful for cases unable to observe all datasets and only able to get its causal evidences, and can be applied in many different data restoring fields [2], e.g. multiple DNA sequence alignment, protein secondary structure prediction [3] and speech recognition. Datasets being used in HMM modelling should be time-serial. For example, speech recognition has temporal structure consisted of statistical models and can be encoded as a sequence of spectral vectors spanning the audio frequency range [4].

Likewise, we'd like to use temporal human action sensor data to make Human Action Recognition system (HAR) using HMM. We've trained our model with Human Activities and Postural Transitions datasets (HAPT) produced by Jorge L, *et al.* [5]. Measurement was done by 30 volunteers carrying a waist-mounted smart phone with embedded inertial sensors. Dataset includes 61 experiments with 30 participants and measured in 50Hz frequency and about 400s duration per experiment. Each experiment contains 12 types of human actions like standing, walking down, *etc.* We separated into 12 part of data so we can easily train our model by action type. Eventually, we've got 1,214 examples and randomly split as train and test dataset as 9:1 ratio respectively.

2. Related Works

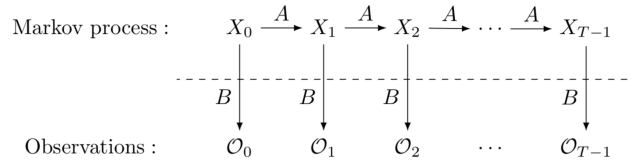
Human activity recognition is nowadays an active research field which aims to understand human behaviour and there have been lots of researches to figure out human behaviours. One of the approaches was method consisted of support vector machines (SVMs) and temporal filters of activity probability estimations within a limited time window [6]. Probability estimation after feature extraction was done

by MAP filtering.

Rubén San-Segundo, *et al.* have implemented system comprising of three main modules: feature extraction, HMMs training and activity recognition. In the training module, 6 HMM modules were being trained to classify human actions with datasets of only six different physical activity types: walking, walking-upstairs, walking-downstairs, sitting, standing and lying down. Due to the lack amount of activity types they've obtained the recognition error rate of 2.5% [7].

3. Preliminaries

3.1. Hidden Markov Model



A hidden Markov model (HMM) allows us to talk about both observed events (like words that we see in the input) and hidden events (like part-of-speech tags) that we think of as causal factors in our probabilistic model. HMM model λ contains transition probability matrix A , a set of state variables X , a sequence of observations O , emission probabilities B , and an initial probability distribution over states π .

We can characterize HMM into 3 fundamental problems;

1) Likelihood: given the model λ , find likelihood $P(O|\lambda)$ whether the observation O is a probable sequence based on λ .

2) Decoding: given an observation sequence O and an model λ , discover the best hidden state sequence X .

3) Learning: given an observation sequence O and the set of states in λ , learn the parameters A and B .

3.2. Forward Algorithm

1. Initialization:

$$\alpha_1(j) = \pi_j b_j(o_1) \quad 1 \leq j \leq N$$

2. Recursion:

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

We use algorithm called the forward algorithm to compute likelihood of observed events. It is an algorithm that uses a table to store intermediate values as it builds up the probability of the observation sequence. The forward algorithm computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence [8].

3.3. Baum-Welch Algorithm

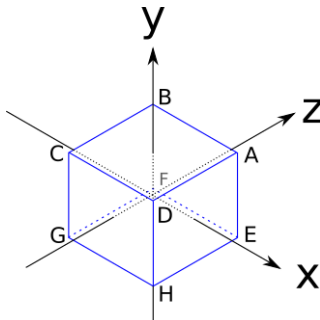
The standard algorithm for HMM training is the forward-backward, or Baum-Welch algorithm, a special case of the Expectation-Maximization algorithm (EM). The algorithm will let us train both the transition probabilities A and the emission probabilities B of model λ .

4. Proposed Method

This section describes overall flows of our proposed method, `jerk8vel8iter50`. Feature extraction and learning with multinomial HMM are the key concepts of our model.

4.1. Feature extraction

Since HAR datasets are given as 3 dimensional vector sets, we have to figure out what feature metric would describe our hidden states of model. We'd like to transform 3 dimensional vectors of acceleration and angular speed into a single metric, so we can train these observations to our HMM model. Figure below shows how we can classify vectors into 8 spaces, A to H, and in number, 0 to 7 w.r.t.



From acceleration observation data, we can apparently determine the direction of jerk, time derivative of acceleration by subtracting neighbouring values [9].

$$\vec{j}(t) = \frac{d\vec{a}(t)}{dt} = \frac{d^2\vec{v}(t)}{dt^2} = \frac{d^3\vec{r}(t)}{dt^3}$$

Inferring from the name of our proposed model `jerk8vel8iter50`, it has 8 jerk states and 8 of those in angular velocity, which is simply determined by signs of vector coordinates. Feature metric is only the measure multiply with these two. For instances, if signs of jerk and angular velocity are $(+, -, +)$ and $(-, -, +)$ each, the measure will be 20 because the state of jerk would be 4(E) and that of angular velocity would be 5(F).

4.2. Multinomial HMM

Using model selection package in sklearn [10], original set was split into train and test data. Every data entity has its feature metric which will be the observation input to HMM model λ . We differentiated into 12 models by activity types and each model learns corresponding activity and classifies test data set with one vs rest method (OVR) with maximum likelihood.

As observation values are all in discrete number, Multinomial HMM from `hmmlearn` was selected and values of hyper-parameters were determined through various empirical approaches. We've finally got `jerk8vel8iter50` model with 00.00% test accuracy and 00000s total training time. Figures below show a table of 12 models' training time, and some of examples when estimates went wrong.

5. Empirical Analysis

Here we describes how we came up with `jerk8vel8iter50` model.

5.1. Feature Metric Approach

When deciding what feature metric would be suitable, we've compared many different models trained for 1 iteration and chose the metric of the model with best test accuracy.

5.2. Iteration Approach

6. Conclusion

References

- [1] ratgo's blog
- [2] Wikipedia, Hidden Markov model
- [3] Byung-Jun Yoon, "Hidden Markov Models and their Applications in Biological Sequence Analysis"

- [4] M. Gales and S. Young “The Application of Hidden Markov Models in Speech Recognition”. *Foundations and Trends in Signal Processing, Vol.1, No.3*, 2007
- [5] Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto and Xavier Parra, “Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set” , 2015
- [6] Jorge L, *et al.* “Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions” *ICANN 2014, LNCS 8681, pp.177–185*, 2014
- [7] Rubén San-Segundo, Julián David Echeverry-Correa, Christian Salamea, José Manuel Pardo “Human activity monitoring based on hidden Markov models using a smartphone” *IEEE Instrumentation and Measurement Magazine*, 2016
- [8] Daniel Jurafsky and James H. Martin. “Hidden Markov Models” *Speech and Language Processing*, 2019
- [9] Wikipedia, Jerk (physics)
- [10] API Documentation of sklearn, model selection, train test split