*Proceedings of the 12th IASTED International Conference*
*Biomedical Engineering (BioMed 2016)*
*February 15 - 16, 2016 Innsbruck, Austria*

# Learning from Life-logging Data by Hybrid HMM: A Case Study on Active States Prediction

Ji Ni, Tryphone Lambrou, Xujiong Ye
School of Computer Science, University of Lincoln
Brayford Pool, Lincoln, LN6 7TS, UK.
{jni, tlambrou, xye}@lincoln.ac.uk

## ABSTRACT

In this paper, we have proposed employing a hybrid classifier-hidden Markov model (HMM) as a supervised learning approach to recognize daily active states from sequential life-logging data collected from wearable sensors. We generate synthetic data from real dataset to cope with noise and incompleteness for training purpose and, in conjunction with HMM, propose using a multiobjective genetic programming (MOGP) classifier in comparison of the support vector machine (SVM) with variant kernels. We demonstrate that the system with either algorithm works effectively to recognize personal active states regarding medical reference. We also illustrate that MOGP yields generally better results than SVM without requiring an *ad hoc* kernel.

## KEY WORDS

eHealth, Machine Learning, Wearable Sensor, Life-logging Data.

## 1. Introduction

Life-logging data collected by wearable sensors has drawn great attention to monitoring people's daily activities for healthcare, commercial and a range of other purposes [1]. In healthcare, there are various responsive instruments for assessment of quality of life (QoL), or for quantifying functional impairment related to vision. These instruments consist of questionnaires, e.g., VF-14 [2], SF-12 [3], in which most questions require a single answer from multiple choices. In order to avoid the subjective bias from the patients and to be time efficient for both patients and the healthcare institute, the personal life-logging data can be collected by mobile sensors, transmitted to the server over internet, and processed and modelled to fulfil questionnaires virtually automatically.

One of the most common questions is to assess people's daily activity state to evaluate his/her general health. For instance, the SF-12 questionnaire concerns a score from 1 to 5 while the conventional study and the public health guide only provides a statistical threshold to output a binary result [4]. Pekka Siirtola *et al.*, recently used machine learning techniques over datasets consisting of 595 people and 678 features to detect sedentary young men with a ternary output [5]. This work employed several multiclass classifiers to model stationary datasets that ignores no sequential dependency if applied to life-logging data. To cope with these limitations, we have applied hidden Markov models (HMM) with medical reference to process the sequential data and yield a multi-state output. In this study, we consider the daily active states as a general health indicator for the question arising from the SF-12 questionnaire, therefore a five-state output is adopted.

Hidden Markov models reach a wide success in modelling sequential data, for instance, in bioinformatics, natural language processing, and etc. It is a probabilistic description of a series of $M$ variant observations emitted from $K$ variant hidden states. The sequence of hidden states is a Markov chain with the probability distribution of each element being conditional on its current and past states. Using HMM for supervised learning tasks usually imposes strong assumptions, e.g., Gaussian mixture model (GMM), for continuous input or Bernoulli for categorical for estimating the emission matrix. When input is a high dimensional continuous vector, we have to trade-off the strength of assumptions against the number of parameters to be optimized. Further, the expectation maximization (EM) algorithm yields only *local optima* that highly depends on initialization.

Common methods to search for an global optima of HMM parameters involves a range of parametric evolutionary algorithms, inclusive of (MO) genetic algorithms [6, 7], particle swarm optimisation [8], etc. However, these methods cannot fundamentally solve the problem when the input space is large. For instance, high dimensional continuous input vectors leads to large sized HMMs even if we have adopted the GMM that parameterizes continuous distribution of input vectors. To address a fair optima in large and sparse space remains a challenging optimisation problem. In order to construct a compact HMM and eliminate the assumptions used in GMM, a hybrid classifier/HMM is adopted to transfer the multidimensional continuous input vectors from the original space to a finite and discrete class space, which effectively reduces the structural complexity of HMM. Not only does the system have no impose of mixture models, it also avoids the initialization dependency and local optima yielded from the EM algorithm. Compared to conventional HMM, superior results are received by employing the hybrid system using SVM classifier in a range of problems [9, 10]. Thus, we propose to employ this system in predicting daily states. Further, we propose

*DOI: 10.2316/P.2016.832-019*

using multiobjective genetic programming (MOGP) as a comparison to illustrate that our algorithm is generally better than the existing SVM/HMM in accuracy and generality, since no *ad hoc* selection of kernels is needed.

This paper presents a scheme for supervised learning from life-logging data using a classifier/HMM. Section 2 describes technical details in classifier/HMM scheme and in generating synthetic data to cope with noise and incomplete data. Section 3 presents competitive results from both MOGP/- and SVM/HMM, followed by the conclusion in Section 4.

## 2. Methodology

### 2.1 Hybrid Classifier/Hidden Markov Model

The general framework is presented in Figure 1.

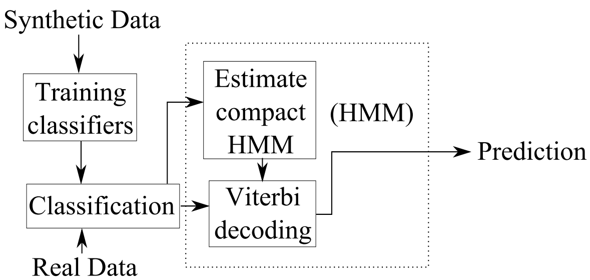


Figure 1. General framework of classifier/HMM system for daily active states prediction.

The synthetic data is generated upon characteristics of real datasets, the detail of which will be presented in Section 2.3. With equal prior of hidden states, the synthetic data copes with noise and imbalance and is used for training classifiers. The real data is then classified with a class label belonging to the class space. Real data is then partitioned into two parts; one is used to estimate the HMM empirically while the other is predicted by Viterbi decoding [11] for assessment purpose.

### 2.2 MOGP and SVM Classifiers

Genetic programming (GP) is a non-parametric evolutionary optimization algorithm using tree-based syntax to represent non-parametric models as shown in Figure 2. For classification models, GP is used to search for a discriminative functions in a rich model space as there is no predetermine structure. The potential benefit is in turn to yield better results for general applications.

The discriminative classifier $f(x) = t_0$ trained by GP is used to classify an input $x_i$ to class "0" if $f(x_i) < t_0$ or class "1" if $f(x_i) \geq t_0$. So, GP models maps the input vector into a scalar on decision space $t \in \mathcal{R}$, where $\mathcal{R}$ is the real domain. A threshold $t_0$ is then determined where the empirical error is minimum. For instance, Figure 2 represents a non-parametric discriminative classifier

$$-(x_1 + 0.34) \cdot x_2 \begin{cases} \geq t_0, decide\ class\ 1. \\ < t_0, decide\ class\ 0. \end{cases}$$

To minimize the empirical error only, however, lead to overfitting models which yield small training error but large test errors. In order to cope with this inherent issue of empirical modelling, we have employed a multiobjective mechanism to minimize the model complexity simultaneously which effectively implements Occam's Razer that suppresses the overfitting and enhance the model generalization.

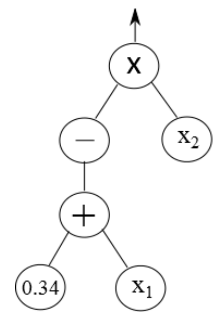


Figure 2. An example of GP tree

The GP parameters are summarized in Table 1. We have run up to 80 000 tree evaluations, each of which newly generates a model. Another termination criterion is 0/1 loss reaches zero.

SVM classifier dominates the classification field due to its solid mathematical background – the statistical learning theory. By minimising the hinge loss [11], SVM converges to a maximal margin classifier with lowest expected risk in the kernel space, which, in turn, achieves low generalization error. In this paper, we have adopted ν-SVM algorithm [12], where ν ∈ [0,1] and is easier for fine-tuning than C-SVM. We have also examined three commonly used kernels, which are radial basis, polynomial and sigmoid to compare with MOGP. For both SVM and MOGP, we basically adopt one-vs-all scheme for multiclass classification.

Table 1
GP parameters

| | |
|---|---|
| Population size | 100 |
| Evolutionary strategy | Steady-state [13] |
| Initialization | Ramped [14];<br>30 repetitions |
| Termination criterion | 80 000 evaluation, or<br>0/1 loss = 0 |
| Crossover<br>Mutation | Point crossover [14]<br>Point mutation [14]; Tree depth = 4 |
| Node Type | Unary minus<br>Addition, Subtraction<br>Multiplication<br>Analytic quotient [15] |

### 2.3 Data Preprocessing

The datasets were collected by "moves-app", a cell phone app using accelerometer and GPS in the handset. When the phone is carried along with the users, the app processes the signals from the accelerometer to recognize walking, running, or moving with transportations. The data is then transmitted to the server through internet. The dataset is a sequence of vectors, each of which consists of people's daily accomplishment of distance, duration and number of steps from his/her physical activities, like walking or running. There are totally 10 datasets collected from 10 people for the experiments. The size of each dataset ranges from 118 days to 401.

In terms of daily active level, there can be a huge difference from person to person. One who favors sports will always have highly active patterns which are not easy to be observed from those inactive people. Thus, the imbalanced real data is not ideal for training classifiers. To solve this problem, we have used the characteristics of the real data to estimate and extrapolate the synthetic data for personalized classifier training purpose.

Recall the input vector, it consists of step, duration, and distance, all of which are highly correlated. We compute the statistical characteristics of speed $S$ and step frequency $F$ for each individual person over all his/her real data and use them as the basis to generate the synthetic datasets. We firstly randomly uniformly generate the duration $dr_i$. The step and distance will be generated by

$$step_i = dr_i \cdot \mathcal{N}(\mu_F, \sigma_F^2)$$
$$distance_i = dr_i \cdot \mathcal{N}(\mu_S, \sigma_S^2)$$

, where $\mathcal{N}$ is Gaussian distribution with its mean $\mu$, variance $\sigma^2$. Training set for each person comprises 1000 data with 200 data per state. Figure 3 shows a sample of synthetic data in five-colour compared with real data presented in black dots. The synthetic data provide a practical simulation to the real data and successfully cope with the issue of imbalance.

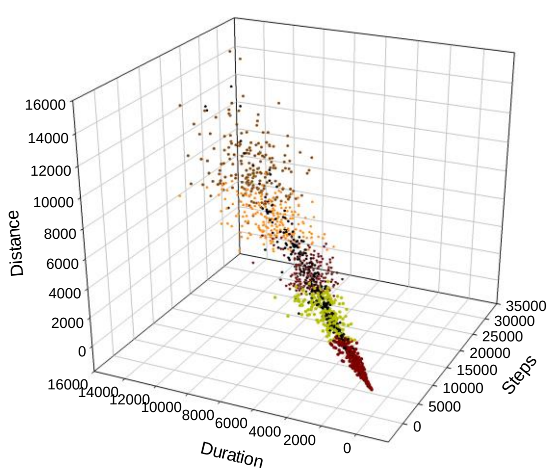


Figure 3. An example of synthetic and real data distribution.

To tag the ground truth for each datum, we have employed 10 000 steps, 60 minutes, and 8km as the *median* (state 3 out of 5) active reference according to the medical literature [4, 16]. We then use linear model to label all five states.

$$Tag_i = \frac{1}{3}\left(\frac{distance_i}{8km} \times 3 + \frac{dr_i}{60min} \times 3 + \frac{step_i}{10000} \times 3\right)$$

, where $Tag_i$ is capped by 5. We have to emphasize that to use a more complex (e.g., non-linear) medical model $f$ that tags the input vector $x_i$, namely $Tag_i = f(x_i)$ is beyond the scope of this paper, but any tagging procedure employed is *independent* to the learning process. We employ clean synthetic data set for training classifiers. For the noisy real data, noise model is typically unknown or dependent on the sensors. We thus assume a typical upper bound of relative error of 10%, without losing generality, and investigate all noise occasions below that. Namely, we assume a Gaussian noise $\mathcal{N}(\mu, (k\mu)^2)$, where $k \leq 0.1$. We examine six points which are *k = {0, 0.02, 0.04, 0.06, 0.08, 0.1}.*

### 3. Experimental Results and Discussion

We investigate the relative performance on four algorithms. In classifier training process, we have employed a 5-fold cross-validation (CV) on the training data for SVM algorithms to fine-tune and select the best parameters. Since MOGP are evolving different models and has no straightforward way for CV, we thus repeat 5 times with random partition of the 1000 synthetic data into training and selecting datasets and select the best results.

For all of the four algorithms in the HMM process, we have 10 repetitions that is similar to leave-one-out 10-fold method. In each fold, we empirically estimate the parameters in HMM using 9/10 of real data, followed by employing Viterbi decoding to perform the final prediction over the whole real data. The average of 10 repetitions (folds) is the mean with respect to variant empirical estimations of HMM parameters and is used to represent the performance of each algorithm per data setting. This method is repeated over 6 noise settings of real data to investigate the robustness and generality of the algorithm over a real noise range.

We have investigated four algorithms by 10 people by 6 noise settings in real data which summed up to 240-element results in total. For the sake of briefness, we select a typical result on Person 1 and present it in Table II to illustrate how we process the data and assess the performance.

Table 2 presents a typical result from an individual person. "HmmG", "HmmR", "HmmP", and "HmmS" represent MO**G**P/HMM and SVM/HMM with **R**adial basis, **P**olynomial, and **S**igmoid kernels, accordingly. On the left side of the table 2, the test error of each algorithm is shown in each column, each row of which regards various noise settings. We notice that the performance of each algorithm is worsened with increasing noise. To directly average test error over all noise settings is unfair as errors from large noise dominate the final result. So, we assign ranks in each noise setting to represent a relative performance of each algorithm. "1" represent the

Table 2
Results of test error and relative ranks over all noise setting using HMM with variant classifiers for Person 1.

| Person_1 | Test Error | | | | Rank | | | |
|---|---|---|---|---|---|---|---|---|
| Noise | HmmG | HmmR | HmmP | HmmS | HmmG | HmmR | HmmP | HmmS |
| 0 | 0.0119 | 0.0085 | 0.0261 | 0.0283 | 2 | 1 | 3 | 4 |
| 0.02 | 0.0205 | 0.0227 | 0.0308 | 0.0368 | 1 | 2 | 3 | 4 |
| 0.04 | 0.0573 | 0.0567 | 0.0463 | 0.0595 | 3 | 2 | 1 | 4 |
| 0.06 | 0.0727 | 0.0765 | 0.0658 | 0.0736 | 2 | 4 | 1 | 3 |
| 0.08 | 0.0894 | 0.0935 | 0.0840 | 0.0900 | 1 | 4 | 2 | 3 |
| 0.10 | 0.1316 | 0.1357 | 0.1265 | 0.1325 | 2 | 4 | 1 | 3 |
| Ranks Expectation (Rank_Exp) | | | | | 1.8333 | 2.8333 | 1.8333 | 3.5 |

best performance, hence the smallest test error while "4" is the worst. In Table 2, the relative ranks are presented on the right half and the rank expectations are summarized in the bottom row. This represents the relative performance expectation over all noise of $k \leq 0.1$ of each algorithm for Person_1.

Table 3
Rank Expectation for all datasets using HMM with variant classifiers

| Rank_Exp | HmmG | HmmR | HmmP | HmmS |
|---|---|---|---|---|
| P1 | 1.8333 | 2.8333 | 1.8333 | 3.5 |
| P2 | 2.8333 | 2.8333 | 1.5 | 2.8333 |
| P3 | 2.25 | 3.1667 | 1 | 3.5833 |
| P4 | 2.5 | 2.5 | 2.5 | 2.5 |
| P5 | 3.25 | 1.5833 | 2.1667 | 3 |
| P6 | 2.5 | 1.6667 | 2.1667 | 3.6667 |
| P7 | 1.25 | 2.0833 | 3.8333 | 2.8333 |
| P8 | 2.5 | 2.5 | 2.5 | 2.5 |
| P9 | 2.1667 | 3.75 | 1.5833 | 2.5 |
| P10 | 2.25 | 2.9167 | 2.4167 | 3.8333 |
| Average of Rank_exp | 2.3333 | 2.5833 | 2.15 | 2.9333 |

We have repeated this procedure over all of the datasets and summarize in the Table 3 that presents rank expectation from each algorithm and over all people. The bottom row shows the expectation of relative performance over all noise settings and over all datasets from each algorithm. We notice that MOGP/HMM yields better results than two SVM/HMM kernels but is interior to polynomial kernels. It indicates that MOGP/HMM is expected to be generally better than SVM/HMM for any people and any sensor noise less than or equal to 0.1, relatively.

Overall, classifier/HMM system with MOGP or SVM have both receive practical results for daily active states prediction as shown in the left half of Table 2. The performance of each algorithm is positively proportional to the noise of test data. However, as long as the relative noise is less than 0.1, the error of any algorithm is always less than 0.14. In most cases, errors are less than 0.1. According to Table 3, SVM/HMM with polynomial kernel obtained highest (smallest) averaged rank of 2.15, followed by MOGP/HMM 2.33. SVM/HMM with radial and sigmoid kernels are ranked worse than MOGP/HMM at 2.58 and 2.93, respectively. The relative performance of the SVM/HMM algorithms varying on the kernel selection illustrates the generality of MOGP/HMM that requires no *ad hoc* kernel functions. The relative performance of MOGP/HMM is typically superior to SVM/HMM considering the commonly used kernels we have tested.

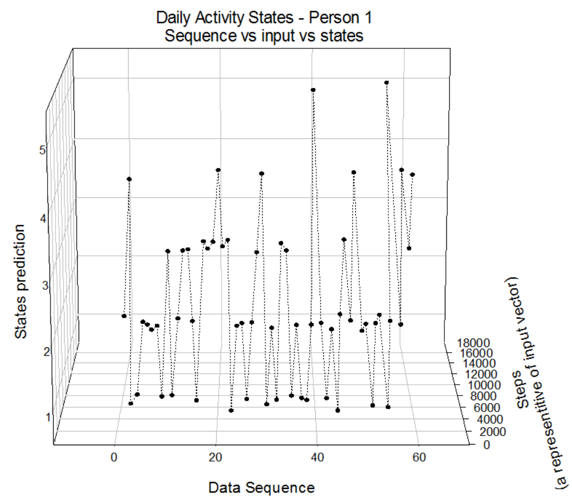


Figure 4(a). A visualization of sequential input vs output; front view

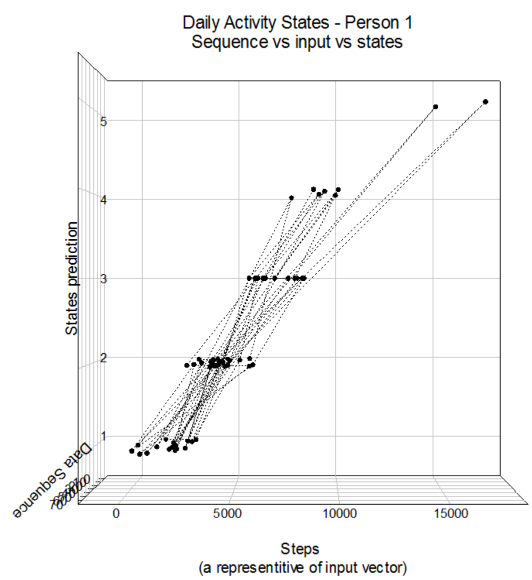


Figure 4(b). A visualization of sequential input vs output; side view

We further visualize the output of our systems. Figure 4(a) shows a front view of a 3D plot of data sequence consisting of 60 elements, each of which consists of sequence number, steps, and states predictions. We only use steps as a representative of the input for visualization purposes. In the sequence, the active state of each day is fluctuating, ranging from 1 to 5. When we have the side view for the same plot, we have Figure 4(b). We find out that the active state is basically positively correlates to the steps, although there are overlaps. This is due to the MOGP/HMM system considers duration, distance and the activity of its neighbourhood as well to yield the final decision. Overall, this system works effectively for daily active states and is potential for general serial supervise learning for healthcare.

## 4. Conclusion and Future Work

In this paper, we have proposed using a hybrid classifier/HMM system for learning from sequential life-logging data and in detail study its performance in human daily active states predictions. We have proposed a MOGP/HMM system, which yield generally better results in comparison of SVM/HMM.

Our current work using HMM is based on first-order Markov assumption that current state depends on one of its previous states which is unnecessarily the reality. Higher-order Markov assumption considering information from more of the previous states will be further investigated to improve the performance.

**Acknowledgement**

This work is supported by the EPSRC grant EP/L023679/1 and European FP7 collaborative project "MyHealthAvatar" (GA No: 600929).

## REFERENCES

[1] C. Gurrin, *et al.*, "Lifelogging: Personal big data," *Foundations and Trends in Information Retrieval*, vol. 8, pp. 1-125, 2014.

[2] C. B. Terwee, *et al.*, "Development of a disease specific quality of life questionnaire for patients with Graves' ophthalmopathy: the GO-QOL," *British Journal of Ophthalmology*, vol. 82, pp. 773-779, 1998.

[3] B. Gandek, *et al.*, "Cross-validation of item selection and scoring for the SF-12 Health Survey in nine countries: results from the IQOLA Project," *Journal of Clinical Epidemiology*, vol. 51, pp. 1171-1178, 1998.

[4] G. C. Le-Masurier, *et al.*, "Accumulating 10,000 steps: does this meet current physical activity guidelines?," *Research Quarterly for Exercise and Sport*, vol. 74, pp. 389-394, 2003.

[5] P. Siirtola, *et al.*, "Detecting and profiling sedentary young men using machine learning algorithms," in *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2014, pp. 296-303.

[6] C. W. Chau, *et al.*, "Optimization of HMM by a genetic algorithm," in *IEEE International Conference onAcoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997.*, 1997, pp. 1727-1730 vol.3.

[7] T. Garcia-Valverde, *et al.*, "Ubiquitous deployment configuration of indoor location services," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1-8.

[8] F. Yang, *et al.*, "Comparison of particle swarm optimization and genetic algorithm for HMM training," in *19th International Conference on Pattern Recognition, 2008. ICPR 2008*, 2008, pp. 1-4.

[9] W. Hao, *et al.*, "A hybrid CSVM-HMM model for acoustic signal classification using a tetrahedral sensor array," in *SENSORS, 2014 IEEE*, 2014, pp. 1352-1355.

[10] A. Ganapathiraju, *et al.*, "Hybrid SVM/HMM architectures for speech recognition," in *INTERSPEECH*, 2000, pp. 504-507.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*: Springer, 2007.

[12] P. H. Chen, *et al.*, "A tutorial on ν-support vector machines," *Applied Stochastic Models in Business and Industry*, vol. 21, pp. 111-136, 2005.

[13] R. Kumar and P. Rockett, "Improved sampling of the Pareto-front in multiobjective genetic optimizations by steady-state evolution: a Pareto converging genetic algorithm," *Evolutionary Computation*, vol. 10, pp. 283-314, 2002.

[14] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[15] J. Ni, *et al.*, "The use of an analytic quotient operator in genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 146-152, 2013.

[16] W. W. K. Hoeger, *et al.*, "One-mile step count at walking and running speeds," *ACSM's Health & Fitness Journal*, vol. 12, pp. 14-9, 2008.