Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 15, No. 3, September 2015, pp. 145-152
http://dx.doi.org/10.5391/IJFIS.2015.15.3.145

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Online Selective-Sample Learning of Hidden Markov Models for Sequence Classification

Minyoung Kim
Department of Electronics & IT Media Engineering, Seoul National University of Science & Technology, Seoul, Korea

ljfis

Abstract

We consider an online selective-sample learning problem for sequence classification, where the goal is to learn a predictive model using a stream of data samples whose class labels can be selectively queried by the algorithm. Given that there is a limit to the total number of queries permitted, the key issue is choosing the most informative and salient samples for their class labels to be queried. Recently, several aggressive selective-sample algorithms have been proposed under a linear model for static (non-sequential) binary classification. We extend the idea to hidden Markov models for multi-class sequence classification by introducing reasonable measures for the novelty and prediction confidence of the incoming sample with respect to the current model, on which the query decision is based. For several sequence classification datasets/tasks in online learning setups, we demonstrate the effectiveness of the proposed approach.

Keywords: Machine learning, Sequence classification, Online learning, Hidden Markov models

1. Introduction

Sequences or time-series data are parts of our everyday life in diverse forms such as videos of image frames, speech signals, financial asset prices and meteorological records, to name a few. Sequence classification involves automatically assigning class labels to these instances in a meaningful way. Owing to the increasing demand for efficient indexing, search, and organization of a huge amount of sequence data, it has recently received significant attention in machine learning, data mining, and related fields. Unlike conventional classification of non-sequential vectorial data, sequence classification entails inherent difficulty originating from potentially variable-length and non-stationary structures.

Recent sequence classification approaches broadly adopt one of the two different frameworks. In the first framework, one estimates the similarity (or distance) measure between pairs of sequences, from which any kernel machines (e.g., support vector machines) or exemplar-based classifiers (e.g., nearest neighbor) can be employed. As the sequence lengths and sampling rates can differ from instance to instance, one often resorts to warping or alignment of sequences (e.g., dynamic time warping), alternatively, incorporating a specially-tailored similarity measure (e.g., spectral or string kernels [1]).

The second framework is based on probabilistic sequence models, essentially aiming to represent an underlying generative model for sequence data. The hidden Markov model

Received: Aug. 14 2015
Revised : Sep. 23, 2015
Accepted: Sep. 24, 2015

Correspondence to: Minyoung Kim
(mikim@seoultech.ac.kr)

(HMM) is considered to be one of the most popular sequence models, and has previously shown broad success in various application areas including automatic speech recognition [2, 3], computer vision [4–6], and bioinformatics [7]. y-based one in several aspects: certain statistical constraints such as Markov dependency structures can be easily imposed. In addition, the model-based approaches are typically computationally less demanding for training, specifically linear time in the number of training samples, while similarity-based methods require quadratic. Thus, throughout the study we focus on HMM-based sequence classification: the details of the model are thoroughly described in Section 2.

Unlike the traditional learning setup where all the labeled training samples are stored and available to a learning algorithm (i.e., *batch learning*), the learning setup we deal with in this study is quite different. We consider the *online selective-sample learning*: it is basically a streaming data scenario where we receive an input sequence (without its class label) one at a time, and the technique is assumed to have no capability of storing the sample for future use. The learning algorithm has an option of either querying the class label or not, and this decision has to be made on the fly based solely on the current data sample (i.e., unable to look into previous samples). The model can then be updated with the sequence sample and the class label (if queried). Of course, there is a limit to the total number of queries that can be made, and therefore the learner's main objective is to select samples for queries that are the most salient and important for learning an accurate model.

Considering the cost of obtaining class labels, which typically requires human experts endeavor, the online learning algorithms can be more favorable. Moreover, they are better suited for various applications, most notably the mobile computing environments, in which there are massive amount of data collected and observed whereas the computing platforms (e.g., mobile devices) have minimal storage and limited computing power. It is worth noting that the online selective-sample learning setup is different from the well-known active learning [8, 9] in machine learning. In the active learning, the algorithm has full access to entire data samples (thus requiring data storage capabilities), and the goal is to output a subset for which the class labels are queried. In this sense, the online selective-sample learning can be more realistic and challenging than the active learning.

Recently there have been attempts to tackle the online selective-sample learning problem. In particular, the main idea of the latest aggressive algorithms (e.g., [10, 11]) is to decide to query labels for samples that appear to be novel (compared to the previous samples) with respect to the current model. Moreover, it is desirable to ask for labels for those data that have less confident class prediction under the current model, which is also intuitively appealing. However, most approaches are limited to vectorial (non-sequential) data and binary classification setups with the underlying classification model confined to the simple linear model.

We extend the idea to the multi-class sequence data classification problem within the HMM classification model. Specifically, we deal with the negative data log-likelihood of the incoming sequence as a measure of data novelty. Furthermore, the entropy of the class conditional distribution given the incoming sequence is considered as a measure of strength/confidence in class prediction. The proposed approach is not only intuitively appealing, but also shown to yield superior performance to the baseline approaches that make queries randomly or greedily. These results have been verified for several sequence classification datasets/problems.

The paper is organized as follows: after introducing a few notations and a formal problem setup, we describe the HMM-based sequence classification model that we deal with throughout the study in Section 2. Our main approach is described in Section 3, and the empirical evaluations are demonstrated in Section 4.

1.1 Notations and Problem Setup

We consider a K-way sequence classification problem, where we let $c \in \{1, ..., K\}$ be the class variable and $\mathbf{x} = \mathbf{x}_1 \dots \mathbf{x}_T$ be the sequence observation (the length T can vary from instance to instance). We assume each time slice $\mathbf{x}_t \in \mathbb{R}^d$ is a real-valued d-dimensional vector.

In the online selective-sample learning setup, the learner receives a sequence one at a time, and decides whether to query its class label or not. More formally, at the i-th stage, the algorithm receives a new sequence $\mathbf{x}^i = \mathbf{x}^i_1 \dots \mathbf{x}^i_{T_i}$ (length T_i), and outputs the class prediction $\hat{c}_i$ for $\mathbf{x}^i$ from the current model. It then decides whether to query the true class label c_i or not, and the learning algorithm may update the classification model using the observed data sample (either $\mathbf{x}^i$ or $(c_i, \mathbf{x}^i)$, latter only when the query is made). In general, there are two (complementary) goals for the learner: to yield an accurate class prediction model and to make as few queries as possible. One possible way of enforcing the goals, which we adopt in this paper, is to introduce the budget B, the upper bound of the

number of queries to be made, and devise an algorithm yielding the smallest classification error with the budget constraint.

2. HMM-based Sequence Classification Model

We consider a probabilistic sequence classification model $P(c, \mathbf{x}) = P(c) \cdot P(\mathbf{x}|c)$, where $P(c)$ is the class prior (modeled by the multinomial distribution over $\{1, \ldots, K\}$), and $P(\mathbf{x}|c)$ is the c-th HMM model (with $c = 1, \ldots, K$) that is responsible for generation of a sequence $\mathbf{x}$ under the class c. In this paper we use the Gaussian-emission HMM models, for which we provide formal descriptions below.

The HMM is composed of two generative components: (i) a (hidden) state sequence $\mathbf{s} = s_1 \ldots s_T$ is generated where each hidden state s_t takes a discrete value from $\{1, ..., S\}$) conforming to the 1st-order Markov dependency, (ii) at each time slice t, the observation $\mathbf{x}_t$ is generated whose distribution (Gaussian) is determined by s_t. More formally, we have the following local conditional models and associated parameters:

$$\begin{aligned} P(s_1 = j) &= \pi_j, \ \ 1 \le j \le S, \\ P(s_t = l | s_{t-1} = j) &= A_{j,l}, \ \ 1 \le j, l \le S, \\ P(\mathbf{x}_t | s_t = j) &= \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \ \ 1 \le j \le S. \end{aligned} \quad (1)$$

The parameters of the HMM are denoted by

$$\boldsymbol{\theta} = \{\pi, A, \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^S\}.$$

In typical cases, the state sequence $\mathbf{s}$ is not observed, and one has to deal with the observation likelihood which is marginalization of the full joint likelihood over all possible hidden sequences, namely $P(\mathbf{x}) = \sum_{\mathbf{s}} P(\mathbf{s}, \mathbf{x})$. This marginalization can be done very efficiently (time linear in sequence length T) using the dynamic programming [12]. Given an HMM model and the observation sequence $\mathbf{x}$, the task of computing the hidden state posteriors (i.e., $P(\mathbf{s}|\mathbf{x})$) is very important. Often referred to as *probabilistic inference*, it can be done efficiently using the forward/backward recursions [12]. Specifically, we denote two key quantities by: $\gamma_t(j) := P(s_t = j|\mathbf{x})$ and $\xi_t(j, l) := P(s_{t-1} = j, s_t = l|\mathbf{x})$.

While the parameter learning of the HMM can typically done by the EM algorithm [13], one can directly optimize the log-likelihood $\log P(\mathbf{x})$ by standard gradient ascent. This is indeed exploited in our online selective-sample learning algorithm since we need to update a model with a single sample within the stochastic gradient optimization framework [14]. The gradient of the observation log-likelihood can be derived as follows:

$$\begin{aligned} \frac{\partial \log P(\mathbf{x})}{\partial \pi_j} &= \frac{\gamma_1(j)}{\pi_j}, \quad \frac{\partial \log P(\mathbf{x})}{\partial A_{j,l}} = \frac{\sum_{t=2}^T \xi_t(j,l)}{A_{j,l}}, \\ \frac{\partial \log P(\mathbf{x})}{\partial \boldsymbol{\Sigma}_j^{-1}} &= \sum_{t=1}^T \frac{\gamma_t(j)\big(\boldsymbol{\Sigma}_j - (\mathbf{x}_t - \boldsymbol{\mu}_j)(\mathbf{x}_t - \boldsymbol{\mu}_j)^\top\big)}{2} \\ \frac{\partial \log P(\mathbf{x})}{\partial \boldsymbol{\mu}_j} &= \sum_{t=1}^T \gamma_t(j) \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_j). \end{aligned} \quad (2)$$

Returning to the classification model, we deal with K HMMs and treat each one as a class conditional density $P(\mathbf{x}|c)$ for each class c. The class prior $P(c)$ is modeled by a multinomial with a K-dimensional paraemter vector p where $p_k = P(c = k)$. Overall the model has parameters $\boldsymbol{\Theta} = \{p, \{\boldsymbol{\theta}^{(k)}\}_{k=1}^K\}$, and $\boldsymbol{\theta}^{(k)} = \{\pi^{(k)}, A^{(k)}, \{\boldsymbol{\mu}_j^{(k)}, \boldsymbol{\Sigma}_j^{(k)}\}_{j=1}^S\}$ is the parameters of the k-th HMM. For simplicity, the hidden state cardinalities (i.e., S) are assumed to be the same across all the component HMMs. We refer to this HMM-based classification model as SC-HMM. In the SC-HMM model, class prediction for an unseen test sample $\mathbf{x} = \mathbf{x}_1 \ldots \mathbf{x}_T$ is done by the maximum-a-posteriori (MAP) decision: $c^* = \arg\max_c P(c|\mathbf{x}) = \arg\max_c P(c, \mathbf{x})$.

3. Online Selective-Sample Learning for SC-HMM

Our online selective-sample learning algorithm for the SC-HMM model is motivated from the aggressive algorithm of [11], and we briefly review their approach here. However, it should be noted that their approach is based on the simple linear model for binary classification of vectorial (non-sequential) data, hence should be differentiated from our extension.

In [11] a linear classification model $y = \text{sgn}(\mathbf{w}^\top \mathbf{z})$ is considered where $\mathbf{z} \in \mathbb{R}^d$ is the input observation vector, $\mathbf{w}$ is the parameter vector of the same dimension, and $\text{sgn}(\cdot)$ returns the sign (± 1) of its argument. The so-called score $\mathbf{w}^\top \mathbf{z}$ is real valued, where its magnitude indicates the confidence in prediction (i.e., a larger score implies that $\mathbf{z}$ lies farther away from the decision boundary, and vice versa). We denote by $\mathbf{z}_i$ the i-th sample received by the algorithm, and by δ_i a binary variable indicating whether i-th sample is queried (1) or not (0).

The model $\mathbf{w}$ is basically estimated by L2-regularized linear regression, which gives rise to the estimate $\mathbf{w} = \mathbf{A}^{-1}\mathbf{b}$ at stage i, where

$$\mathbf{A} = \mathbf{I} + \sum_{j<i} \delta_j \mathbf{z}_j \mathbf{z}_j^\top, \quad \mathbf{b} = \sum_{j<i} \delta_j \mathbf{z}_j y_j. \quad (3)$$

Note that (3) can be computed/updated in an online fashion. Then the model's score is computed as: $p = \mathbf{z}_i^\top \mathbf{A}^{-1}\mathbf{b}$. Whether to query y_i or not is then based on this confidence, namely we query if p is smaller than some threshold, and this decision strategy is employed in [10, 11]. In [11], it is further considered the novelty of the sample $\mathbf{z}_i$, which is measured by $r = \mathbf{z}_i^\top \mathbf{A}^{-1}\mathbf{z}_i$. This is based on the 0-mean Gaussian assumption for $\mathbf{z}$, and we query y_i if r is larger than certain threshold.

In their algorithm, due to the simple linear regression model and Gaussianity assumption, the query decision rule is easily derived and the update equation becomes simple. For sequence classification, the underlying model is a rather complex SC-HMM, and extension of the idea requires further endeavor. At each stage i, upon receiving a new sequence $\mathbf{x}^i$, our online selective-sample learning algorithm performs two main steps: 1) decide whether to query the true class label c_i or not, and 2) update the model with the current sample, either $(c_i, \mathbf{x}^i)$ if queried or $\mathbf{x}^i$ if not. We provide detailed derivations for each step in the subsequent sections.

3.1 Decision for Query

First, decision to query is made if the incoming sample meets either of two conditions with respect to the current model $\boldsymbol{\Theta}$. $(\texttt{Cond-1})$ $-\log P(\mathbf{x}^i;\boldsymbol{\Theta}) \geq \tau_{NLL}$ indicates the current data sample attains high negative log-likelihood, or equivalently, the sample appears to be novel for the current model. The threshold τ_{NLL} is properly chosen. $(\texttt{Cond-2})$ $-H(p(c|\mathbf{x}^i;\boldsymbol{\Theta})) \leq \tau_{NCE}$ implies that the entropy of the posterior class distribution for the current sample is high, meaning that the confidence in prediction is not very strong. Here $H(P(c|\mathbf{x})) = -\sum_{c=1}^{K} P(c|\mathbf{x}) \cdot \log P(c|\mathbf{x})$ is the entropy of the class posterior. Again we choose the threshold parameter τ_{NCE} adequately.

3.2 Model Update

The second step is to update the model $\boldsymbol{\Theta}$ using the current data sample. The true class label c_i is either available or not depending on the decision in the above step, and we let the binary variable δ_i record it (i.e., $\delta_i = 1(0)$ if queried (not)). To present our model update algorithm, we begin with the batch data learning formulation assuming all labeled and unlabeled data can be accessed. We then derive the stochastic gradient update rule from the batch formulation.

In the batch learning case with n data samples, the learning problem can be formulated with partially labeled data (sine we have selectively labeled samples) as follows:

$$\max_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{n} \Big(\delta_i \cdot \log P(c_i, \mathbf{x}^i;\boldsymbol{\Theta}) + \lambda \cdot \log P(\mathbf{x}^i;\boldsymbol{\Theta}) \Big). \quad (4)$$

Here note that we can exploit even the unlabeled samples (the second term) by maximizing the marginal log-likelihood where the two objectives are balanced by the hyperparameter $\lambda \geq 0$.

In the stochastic gradient ascent [14], instead of computing the full gradient of (4), it only updates the model using the gradient for a single sample, say i, that is,

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \eta \cdot \nabla_{\boldsymbol{\Theta}} \Big(\delta_i \cdot \log P(c_i, \mathbf{x}^i) + \lambda \cdot \log P(\mathbf{x}^i) \Big), \quad (5)$$

which can be seen as an unbiased sample for the total gradient. The constant $\eta > 0$ is a learning rate which we fix throughout the iterations (e.g., $\eta = 0.001$).

The nice thing is that (5), although derived from a batch setup, can be computed in an online fashion, and it exactly forms our model update equations. To be more specific to the SC-HMM, the gradients can be computed by (for each $k = 1, \ldots, K$):

$$\begin{aligned} \nabla_{\boldsymbol{\theta}^{(k)}} \log P(c_i, \mathbf{x}^i) &= I(c_i = k) \cdot \nabla_{\boldsymbol{\theta}^{(k)}} \log P(\mathbf{x}^i|k) \\ \nabla_{p_k} \log P(c_i, \mathbf{x}^i) &= I(c_i = k), \end{aligned} \quad (6)$$

where $I(\cdot)$ is the 1/0 indicator function, and the gradient in the latter exactly follows the HMM gradient (2). For the marginal log-likelihood term, since

$$\nabla_{\boldsymbol{\Theta}} \log P(\mathbf{x}) = \mathbb{E}_{P(c|\mathbf{x})}[\nabla_{\boldsymbol{\Theta}} \log P(c, \mathbf{x})],$$

$$\begin{aligned} \nabla_{\boldsymbol{\theta}^{(k)}} \log P(\mathbf{x}^i) &= P(c = k|\mathbf{x}^i) \cdot \nabla_{\boldsymbol{\theta}^{(k)}} \log P(\mathbf{x}^i|k) \\ \nabla_{p_k} \log P(\mathbf{x}^i) &= P(c = k|\mathbf{x}^i). \end{aligned} \quad (7)$$

The initial model is chosen randomly and blindly, thus tends to select first a few samples for query, which is a desirable strategy considering that model is inaccurate with little labeled data received at initial stages. We stop the algorithm when the number of queries made reaches the budget constraint B. The proposed online selective-sample learning algorithm is summarized in Alg. 1.

4. Empirical Evaluation

In this section we evaluate the classification performance of the proposed online selective-sample learning algorithm

Algorithm 1 Online Selective-Sample SC-HMM Learning

Input: Initial SC-HMM model $\boldsymbol{\Theta}$ and the budget B.
Output: Learned model $\boldsymbol{\Theta}$ and the stage-wise class predictions $\hat{c}_1, \hat{c}_2, \ldots$.
Initialize the number of queries made $b = 0$.
Repeat for $i = 1, 2, \ldots,$:
1) Take the incoming sequence $\mathbf{x}^i$.
2) Output class prediction $\hat{c}_i = \arg\max_c P(c|\mathbf{x}^i; \boldsymbol{\Theta})$.
3) Compute $r = -\log P(\mathbf{x}^i; \boldsymbol{\Theta})$.
4) Compute $p = -H(p(c|\mathbf{x}^i; \boldsymbol{\Theta}))$.
5) Query if $b < B$ and ($p \leq \tau_{NCE}$ or $r \geq \tau_{NLL}$).
6) Set $b \leftarrow b + \delta_i$.
7) Update the model $\boldsymbol{\Theta}$ using (5).

on several real-world time-series datasets including human gait/activity recognition and facial emotion prediction. For each dataset we form online learning setups by feeding a learning algorithm one sample at each stage randomly drawn from the data pool, where we restrict the algorithm to make queries up to B times. We test with two different values of budget B: 10% and 30% of the entire data samples. Once the algorithm uses up the budget, from then on it can only exploit the incoming sequences only (without class labels) to update the model.

Our algorithm is compared with two fairly basic online learning strategies: the first is to make queries at the first B stages (greedy approach; denoted by `GREEDY`), and the second one makes random queries (denoted by `RANDOM`). The former is reasonable in the sense that the model is initially incorrect, thus trying to learn with labeled data as early as possible. The related hyperparameters (e.g., the learning rate η and the impact of the marginal log-likelihood term λ in model update) are chosen identical across competing models for fair comparison. As a reference, we also contrast with the online learner that is not restricted by the budget constraint (i.e., it can make query every stage), which perhaps provides an upper bound for the classification accuracy (denoted by `QRYALL`).

As a performance measure, we accumulates the test errors up to n stages. That is, the prediction error of an online algorithm is: $\frac{1}{n}\sum_{i=1}^{n} I(\hat{c}_i \neq c_i)$. We provide detailed descriptions of the datasets, experimental setups, and test results in the subsequent sections.

4.1 Human Gait Recognition

The classification task we deal with is to identify a person based on his/her gait sequence. We collect data from the speed-control human gait database [15, 16], where we focus on samples from 6 different subjects to form a $K = 6$-way multi-class classification setup. Each subject performs walking motion with four different speeds (0.7m/s, 1.0m/s, 1.3m/s, 1.6m/s), and the task is to predict a subject regardless of the walking speed. For the sequence representation, we take 3D motion captures of some marker points at the lower body parts (refer to [15] for details). We then select sub-sequences of lengths around 80 with random starting/ending positions.

Table 1. Accumulated test prediction errors ($\%$) on human gait recognition data

Methods	$B = 0.1n$	$B = 0.3n$
Proposed	31.23 ± 2.62	27.72 ± 2.29
`GREEDY`	38.55 ± 1.96	31.42 ± 4.73
`RANDOM`	45.71 ± 6.33	31.51 ± 2.69
`QRYALL`	19.57 ± 1.39	

For $n = 648$ sequences, we form online learning setups with two budget setups, $B = 0.1n$ and $B = 0.3n$, and it is repeated 5 times to report the average test errors. The results are summarized in Table 1. The reference `QRYALL` that makes unlimited queries, as expected, attains the lower bound for test errors of all competing methods. For both budget setups, the proposed approach consistently outperforms the other two methods, which can be attributed to its more sophisticated decision strategy based on sample novelty and/or model's certainty on class prediction, rather than simple greedy or random querying strategies.

It is also interesting to see that when the budget B increases from 10% to 30%, the difference between `GREEDY` and `RANDOM` becomes small, which can be explained as follows: as we have more labeled samples, it is less critical when the labels are asked. Also, for smaller B, the differences between our method and two competing ones are more significant, which emphasizes the effectiveness of the proposed method under a more restricted budget scenario, namely that when only a few queries are allowed to be made, the careful decision strategy in our proposed approach yields outstanding performance.

4.2 Facial Emotion Classification

Next we consider the problem of recognizing facial emotion from a video sequence comprised of facial image frames that undergo changes in facial expression. From the Cohn-Kanade facial expression video dataset [17], we collect videos of two emotions, fear and happiness. Differentiating these two emo-

Table 2. Accumulated test prediction errors (%) on facial emotion classification data

Methods	$B = 0.1n$	$B = 0.3n$
Proposed	31.51 ± 2.66	24.89 ± 2.47
GREEDY	37.12 ± 4.24	27.77 ± 1.94
RANDOM	46.47 ± 9.43	27.63 ± 1.31
QRYALL	22.01 ± 1.88	

Table 3. Accumulated test prediction errors (%) on human activity recognition data

Methods	$B = 0.1n$	$B = 0.3n$
Proposed	51.56 ± 8.67	47.77 ± 2.33
GREEDY	57.60 ± 1.37	50.11 ± 4.33
RANDOM	62.18 ± 4.75	52.96 ± 5.74
QRYALL	40.73 ± 1.57	

tions are recognized as a hard problem in that visually they are very similar to each other. We tackle this binary classification task.

For the sequence representation, we use certain image features extracted from images as follows. We first estimate the tight bounding boxes for faces using the standard face detector (e.g., Viola and Jones [18]), then normalize the sizes of the cropped face images. Then the Haar-like features are extracted for each frame where we follow the procedure similar to that of [19]. The last step is to reduce the dimensionality by principal component analysis. We obtain $n = 139$ sequences with lengths varying from 10 to 30 for about 90 different subjects, and there are 54 sequences for the fear class and 85 for happiness.

The test errors are depicted in Table 2, where the reference QRYALL again gives lower bound on the test errors. Our proposed method outperforms the competing methods significantly and consistently for all budget setups, while for $B = 0.3n$ it attains error rate nearly close to the lower bound. The results again signify the superiority of the proposed query decision strategy.

4.3 Human Activity Recognition

Last but not least, we tackle the problem of human activity recognition from a sequence of motion localization records. We obtain the dataset from the UCI machine learning repository [20], where the original goal was to reconstruct locations and postures from the localization data recorded from wearable tags at articulation points of subjects performing actions [21]. In this experiment, we focus on the task of recognizing the activity type. For this purpose, we collect from three different subjects about 360 sequences of three activities: *walking*, *lying*, and *sitting*. For the sequence representation, we use the 3-dim velocity information evaluated from 3D coordinates data obtained from the left-ankle tag.

For this 3-way classification problem, we form online learning setups similarly as previous experiments. The test results are summarized in Table 3. We again have similar behavior as the former two experiments. The performance of the proposed approach is outstanding: in particular, it even attains more accurate prediction with the smaller budget constraint. In this dataset, due to the rather limited feature information, the overall prediction performance is low, and under such scenarios, the sophisticated query decision in our algorithm is shown to yield viable solutions.

5. Conclusion

In this paper we have proposed a novel online selective-sample learning algorithm for multi-class sequence classification problems. Under the HMM-based sequence classification model, we devise reasonable criteria for decision for query based on the data novelty (likelihood of the incoming data) and the class prediction confidence (entropy of the class posterior). For several sequence classification datasets/tasks in online learning setups, we have shown that the proposed algorithm yields superior prediction accuracy to greedy or random approaches even with a limited query budget. One current issue may be how to choose the threshold parameters adequately, and we leave it as our future work to investigate a systematic way to tune those.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This study was supported by the Research Program funded by the Seoul National University of Science and Technology.

References

[1] C. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," *Pacific Symposium on Biocomputing*, vol. 7, pp. 566–575, 2002.

[2] B. H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," 1985. AT&T Technical Journal.

[3] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," 2007. Advances in Neural Information Processing Systems 19.

[4] T. Starner and A. Pentland, "Real-time American sign language recognition from video using hidden Markov models," 1995. International Symposium on Computer Vision.

[5] A. D. Wilson and A. F. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[6] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," 2003. Computer Vision and Pattern Recognition.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchenson, *Biological Sequence Analysis*. Cambridge University Press, 2002.

[8] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," 2005. Conference on Learning Theory.

[9] A. Beygelzimer, S. Dasgupta, and J. Langford, "Importance weighted active learning," 2009. International Conference on Machine Learning.

[10] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-case analysis of selective sampling for linear classification," *Journal of Machine Learning Research*, vol. 7, pp. 1205–1230, 2006.

[11] K. Crammer, "Doubly aggressive selective sampling algorithms for classification," 2014. International Conference on AI & Statistics.

[12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 185–197, 1977.

[14] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[15] R. Tanawongsuwan and A. Bobick, "Characteristics of time-distance gait parameters across speeds," 2003. Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, TR GIT-GVU-03-01.

[16] R. Tanawongsuwan and A. Bobick, "Performance analysis of time-distance gait parameters under different speeds," 2003. International Conference on Audio and Video Based Biometric Person Authentication.

[17] J. Lien, T. Kanade, J. Cohn, and C. Li, "Detection, tracking, and classification of action units in facial expression," 1999. Journal of Robotics and Autonomous Systems.

[18] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2001.

[19] P. Yang, Q. Liu, and D. N. Metaxas, "Rankboost with l1 regularization for facial expression recognition and intensity estimation," 2009. International Conference on Computer Vision.

[20] S. Hettich and S. D. Bay, "The UCI KDD Archive (http://kdd.ics.uci.edu)," 1999. Irvine, University of California, Information and Computer Science.

[21] B. Kaluza, V. Mirchevska, E. Dovgan, M. Lustrek, and M. Gams, "An agent-based approach to care in independent living," 2010. International Joint Conference on Ambient Intelligence, Malaga, Spain.

Minyoung Kim received his BS and MS degrees both in Computer Science and Engineering from Seoul National University, South Korea. He earned a PhD degree in Computer Science from Rutgers University in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an Assistant Professor in the Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.