Multiple Observations HMM Learning by Aggregating Ensemble Models

Nazanin Asadi, Abdolreza Mirzaei, and Ehsan Haghshenas

***Abstract*—Hidden Markov Model (HMM) is a widespread statistical model used in cases where the system involves not fully observable data sequences such as temporal pattern recognition and signal processing. The most difficult problem in dealing with HMMs is the training procedure, or parameter learning, for which several approaches has been proposed. Nevertheless, these methods suffer from trapping in local maxima and still no tractable algorithm is present to overcome this problem. On the other hand, good performances of ensemble methods, where multiple models are employed to obtain the target model, lead to considering ensemble learning in the HMM training problem. Until now, just a few ensemble methods have been proposed for HMMs which lack strong theoretical background, or do not involve all the basic models to construct the final model. Hence in this paper a new ensemble learning method for HMMs is proposed which takes advantage of information theory measures, specifically Rényi entropy, and addresses the mentioned problems of previous methods. In agreement with this claim, the results show superiority of the proposed method over other compared methods for both synthetic and real-world datasets. Besides, the proposed ensemble method succeeded to meet the performance of other methods with much lower required training samples.**

***Index Terms*—Hidden Markov model, ensemble, information theory, HMM learning, HMM training.**

I. Introduction

HIDDEN MARKOV MODELs (HMMs) are well-known stochastic models for practical applications which deal with time-varying signals. The main property of these statistical models is their capability in characterizing the signal as a parametric random process as well as estimating the parameters of this stochastic process in a precise well-defined manner. The basic theory of HMM was introduced in the late 1960s and early 1970s followed by studying and implementing its application in speech processing during 1970s [1]. Besides speech recognition, HMM is widely used in various applications like computational biology [2], [3], text categorization [4] and computer vision including gesture recognition [5], face recognition [6], hand writing recognition [7] and image classification [8].

One main part in modeling statistical data with HMM is the training phase. Since 1970 when Baum and his colleagues introduced single observation model training in succession of papers [9]–[13], several training methods have been proposed. In 1977, Dempster, Laird and Rubin introduced the Expectation-Maximization (EM) method for maximum likelihood estimation [14]. Given statistical model observations, consisting of latent variables with unknown parameters, the task of EM is to find the maximum likelihood estimates of parameters. It should be noted in these cases, that the solution of equations cannot be solved directly. Nonetheless, due to the small number of samples obtained by a single sequence observation, the major problem with the trained model is that there is not any reliable estimation for model parameters. Therefore, the problem of multiple observation sequences training arises. Having K observation sequences from the same class, the problem is to estimate the parameters of the corresponding HMM model in order to have a high likelihood of generating all K observation sequences. In 1983, the first maximum likelihood estimation method for HMM multiple observation training was presented by Levinson, Rabiner and Sondhi, by using gradient technique and assuming that all observations are independent of each other [15]. Another way to achieve such a model is Rabiner's method in which at each step of the training process all K observation sequences are used to estimate parameters using Baum-Welch re-estimation procedure [16].

It is generally agreed that ensemble classifier models usually perform better than single classifiers. Mackay was the first person who applied ensemble learning method for solving this problem in 1997 [17]. In his ensemble method for training an HMM, each of the K training observations was separately trained by the Baum-Welch algorithm and then the combination of these models was used to form the final model. Mackay's method was not revisited until Davis *et al.* introduced their ensemble method, which still is the most popular one [18]. Their proposed method is just a simple averaging technique, but in spite of its simpleness it performed really effectively in applications like anomaly intrusion detection [19], [20] and video gesture recognition [21]. In the latter, Davis and Lovell also introduced another ensemble method called Viterbi Path Counting (VPC). It is based on the Viterbi algorithm and upgrades parameters with counting transitions and corresponding emissions in Vitrbi path of each observation sequence. They claimed this training algorithm performs well too, but it is dependent on the order of observation sequences given to the algorithm, and this method can be considered as incremental learning rather than ensemble learning. They also proposed a variation of their first ensemble learning, in which searching for the best relative permutation set for ensemble averaging is also included in the training procedure [22].

Manuscript received October 03, 2012; revised May 14, 2013 and July 13, 2013; accepted August 11, 2013. Date of publication August 29, 2013; date of current version October 16, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrzej Cichocki.

The authors are with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran (e-mail: n.asadi@ec.iut.ac.ir; mirzaei@cc.iut.ac.ir; e.haghshenas@ec.iut.ac.ir).

Digital Object Identifier 10.1109/TSP.2013.2280179

Most of the existing ensemble learning techniques used for HMMs look at HMMs as black boxes and produce the final classification by aggregating only the outputs of the classifiers. In other words, the combination occurs at the decision level using methods like bagging [23] and boosting [24], [25], and does not take into account the internal structure of HMMs. As an example, in [26] Jian *et al.* proposed an ensemble learning for motion recognition and retrieval in which AdaBoost [27] is applied to combine weak HMM learners to form strong learners. However, the computation of such a large ensemble may be slow. In particular, in a resource-limited environment, using a large ensemble is a challenging task. Therefore, proposing a methodology for representing large HMM ensembles in a compact and informative HMM is of high importance. On the other hand, aggregating the HMMs of an ensemble to a single HMM results in robustness of the parameters. Therefore, this method can be viewed as an effective HMM parameter estimation method.

In this paper, due to the above mentioned reasons for the superiority of ensemble learners, with emphasis on retaining the structures of basis learners in forming the final model, a new ensemble method is proposed for Discrete Hidden Markov Models where the final model outperforms previous ensemble methods. The main strength of this method is its information theory basis, and interestingly, as discussed in the future sections, Davis's ensemble learning is a special case of the proposed method.

The remainder of this paper is organized as follows: Section II gives an introduction to HMM notation and parameters along with a describtion of HMM training algorithms, related to this work. In Section III the proposed method is explained in detail and following that, in Section IV, the results of experiments are discussed. Lastly, conclusions come in Section V.

II. Background and Notation

A. Discrete Hidden Markov Model (HMM)

A hidden Markov model is a doubly stochastic process which consists of an underlying process in hidden states and its corresponding observable process. In the discrete hidden Markov model each state has a discrete output probability distribution producing observations. The model is described as $\lambda = (A, B, \Pi)$ and is characterized by the following elements:

- A set of N hidden states:

$$S = \{s_1, s_2, \ldots, s_N\} \tag{1}$$

- A set of M observation symbols:

$$V = \{v_1, v_2, \ldots, v_M\} \tag{2}$$

- Transition matrix $A = \{a_{ij}\},\ 1 \leq i, j \leq N$ where:

$$a_{i,j} = P[q_{t+1} = s_j | q_t = s_i], \tag{3}$$

such that

$$a_{ij} \geq 0, \quad \sum_{j=1}^{N} a_{ij} = 1, \tag{4}$$

is the time-independent probability of having transition to state s_j at step $t+1$ given that the state at time t was s_i (the state at time t is defined by q_t).
This matrix, due to the properties mentioned, is also called the *state transition probability distribution*.

- Emission matrix $B = \{b_{ik}\}, 1 \leq i \leq N, 1 \leq k \leq M$ where:

$$b_{ik} = P[v_k \text{ at } t | q_t = s_i] \tag{5}$$

such that

$$b_{ik} \geq 0, \quad \sum_{k=1}^{M} b_{ik} = 1, \tag{6}$$

is the probability of observing symbol v_k at state s_i at time t.
Similar to the transition matrix, this matrix is also named as *observation symbol probability distribution*.

- Initial state probability distribution $\Pi = \{\pi_i\}, 1 \leq i \leq N$ where:

$$\pi_i = P[q_1 = s_i], \tag{7}$$

such that

$$\pi_i \geq 0, \quad \sum_{k=1}^{N} \pi_i = 1, \tag{8}$$

is the probability of having s_i as the starting state.

B. Model Training

Baum-Welch algorithm is an Expectation-Maximization (EM) technique, generally used to train an HMM model by re-estimating the probabilities of the transition and emission matrices [1]. This re-estimation procedure starts with initiating the parameters randomly and updates them to obtain a new model which has a higher probability of producing the given sequence iteratively. This procedure continues until the local maximum is reached, meaning no more improvement would be achieved by generating a new model from the current one. However, this algorithm is proposed for single sequence training.

1) Rabiner Multiple Sequence Method: Assuming K observation sequences generated from the same model, Rabiner's method tries to find a model which best describes the generated sequences. At each step of this method all of the K training sequences are used to re-estimate the parameters simultaneously [16]. The iterative formula which is used is as follows:

$$\bar{a}_{ij} = \frac{\sum_k W_k \sum_{t=1}^{T_k} \alpha_i^{(k)} a_{ij} b_{jO_{t+1}^{(k)}} \beta_{t+1}^{(k)}(j)}{\sum_k W_k \sum_{t=1}^{T_k} a_t^{(k)}(i) \beta_t^{(k)}(i)} \tag{9}$$

$$\bar{b}_{ij} = \frac{\sum_k W_k \sum_{O_t^{(k)} = v_j} \alpha_t^{(k)}(i) \beta_t^{(k)}(i)}{\sum_k W_k \sum_{t=1}^{T_k} a_t^{(k)}(i) \beta_t^{(k)}(i)}, \tag{10}$$

where $W_k = 1/P_k, k \in [1 \dots K]$ and P_k is the probability of estimating the kth sequence by the current model calculated by the Forward algorithm. The parameters $\alpha_t^{(k)}(i)$ and $\beta_t^{(k)}(i)$ are the same as the ones used in Forward and Backward evaluation algorithms in [1].

2) Ensemble-Average Learning for HMM: Another method for multiple observation sequence training is the technique introduced by Davis *et al.*, which find models with significantly higher likelihood than Rabiner's method [18], [28], [22]. This method is a special case of the method suggested by Mackay [17]. In this ensemble learning, the Baum-Welch re-estimation procedure is run separately on each observation, and the final model is produced by combining these trained models. Davis investigated several weighted average techniques for combination while among all of them simple averaging revealed best performance [18]. The parameters in simple averaging are in the following form:

$$\bar{a}_{ij} = \frac{\sum_k W_k a_{ij}^{(k)}}{\sum_k W_k}, \tag{11}$$

$$\bar{b}_{ij} = \frac{\sum_k W_k b_{ik}^{(k)}}{\sum_k W_k}, \tag{12}$$

$$\bar{\pi}_i = \frac{\sum_k W_k \pi_i^{(k)}}{\sum_k W_k}. \tag{13}$$

where W_k is the weighting factor of kth sequence and $\lambda^{(k)} = (A^{(k)}, B^{(k)}, \pi^{(k)})$, the generated model for that sequence.

III. Methodology

As described before, the aim of our ensemble method is to obtain a model which is a good representative of all the given models. If we look at HMM parameters, including prior, transition, and emission matrices as descriptors matrix, the desirable final model descriptors matrix should try to have the minimum distance from basic models. Since rows of these descriptors are probability distributions, information-theoretic measures can be used for characterizing their affinity. To calculate the similarity between two probability distribution functions, several divergence measures, in the form of distance function, have been proposed [29], [30].

Let the number of basic matrices be L. p^k represents the kth matrix and p_i^k and p_i^* are the ith row of p^k and aggregated matrix (p^*), respectively. Assuming the divergence measure is D, the average distance of p_i^* from corresponding rows of basic distributions is:

$$D(i) = \frac{1}{L}\sum_{k=1}^{L} D(p_i^k, p_i^*). \tag{14}$$

For p_i^* to be a good representer of all $p_i^k, 1 \le k \le L$, it should minimize (14). In the proposed ensemble method of this paper Rényi divergence measure is used. In the following section, we provide further information about Rényi divergence measure.

A. Combination Operator

In information theory, the Rényi entropy is a generalization of Shannon entropy for quantifying the diversity, uncertainty or randomness of a system [29]. The Rényi entropy of order α is defined as:

$$H_{R\alpha}(P) = \frac{1}{1-\alpha} log\left(\sum_{j=1}^{n} p_j^{\alpha}\right), \tag{15}$$

where P is a discrete random variable and $P_j = \Pr\{X = x_j\}$. The Rényi divergence of order α from a distribution P to a distribution Q is defined to be:

$$\begin{aligned} RD_{\alpha}(P,Q) &= \frac{1}{\alpha-1} log\left(\sum_{j=1}^{n} \frac{p_j^{\alpha}}{q_j^{\alpha-1}}\right) \\ &= \frac{1}{\alpha-1} log\left(\sum_{j=1}^{n} p_j^{\alpha} q_j^{1-\alpha}\right). \end{aligned} \tag{16}$$

By substituting (16) into (14) we obtain:

$$D(i) = \frac{1}{L}\sum_{k=1}^{L} RD_{\alpha}\left(p_i^k, p_i^*\right). \tag{17}$$

Since $D(i)$ represents the distance, finding the p_i^* which minimizes this value is desired. Assuming p_i^* as a probability distribution function leads to have:

$$\sum_{j=1}^{N} p_{ij}^* = 1,\ p_{ij}^* \ge 0. \tag{18}$$

In order to add constraint (18) to minimization problem of (17), the Lagrange multiplier is used:

$$D'(i) = D(i) + \lambda\left(1 - \sum_{j=1}^{N} p_{ij}^*\right). \tag{19}$$

The minimum amount of the above term is obtained by setting its derivative with respect to p_{im}^* equal to zero, i.e., $\frac{\partial}{\partial p_{im}^*} D'(i) = 0$. Here, p_{im}^* is the variable which minimizes the term. Derivative of (19) with respect to p_{im}^* is:

$$\begin{aligned} \frac{\partial}{\partial p_{im}^*} D'(i) &= \frac{1}{L(\alpha-1)}\sum_{k=1}^{L} \frac{\partial}{\partial p_{im}^*}\left[log\sum_{j=1}^{n}(p_{ij}^*)^{\alpha}(p_{ij}^k)^{1-\alpha}\right] - \lambda \\ &= \frac{\alpha}{L(\alpha-1)ln2}\sum_{k=1}^{L}\frac{(p_{im}^*)^{\alpha-1}(p_{im}^k)^{1-\alpha}}{\sum_{j=1}^{n}(p_{ij}^*)^{\alpha}(p_{ij}^k)^{1-\alpha}} - \lambda. \end{aligned} \tag{20}$$

By setting (20) equal to zero and calculating the value of p_{im}^*, any value of p_{im}^* depends on any other $p_{im'}^*, m' \ne m$. If, for simplifying the above term, we assume $log\sum_{j=1}^{n}(p_{ij}^*)^{\alpha}(p_{ij}^k)^{1-\alpha} = 1$, the value of p_{im}^* can be

computed regardless of the values of any other $p^*_{im'}, m' \neq m$. Thereby:

$$\frac{\partial}{\partial p^*_{im}} D'(i) = \frac{\alpha (p^*_{im})^{\alpha-1}}{L(\alpha-1)ln2} \sum_{k=1}^{L} (p^k_{im})^{1-\alpha} - \lambda. \tag{21}$$

Solving (21) for p^*_{im} we have:

$$\frac{\partial}{\partial p^*_{im}} D'(i) = 0 \tag{22}$$

which leads to

$$P^*_{im} = \left[\frac{\lambda L(\alpha-1)ln2}{\alpha \sum_{k=1}^{L} (p^k_{im})^{1-\alpha}} \right]^{\frac{1}{\alpha-1}}. \tag{23}$$

By substituting (22) in $\sum_{j=1}^{n} p^*_{ij} = 1$, λ is obtained as below:

$$1 = \sum_{j=1}^{n} p^*_{ij} = \sum_{j=1}^{n} \left[\frac{\lambda L(\alpha-1)ln2}{\alpha \sum_{k=1}^{L} (p^k_{ij})^{1-\alpha}} \right]^{\frac{1}{\alpha-1}}, \tag{24}$$

or

$$1 = \left(\frac{\lambda L(\alpha-1)ln2}{\alpha} \right)^{\frac{1}{\alpha-1}} \sum_{j=1}^{n} \left[\frac{1}{\sum_{k=1}^{L} (p^k_{ij})^{1-\alpha}} \right]^{\frac{1}{\alpha-1}} \tag{25}$$

and as a result:

$$\lambda^{\frac{1}{\alpha-1}} = \left(\frac{\alpha}{L(\alpha-1)ln2} \right)^{\frac{1}{\alpha-1}} \sum_{j=1}^{n} \left[\frac{1}{\sum_{k=1}^{L} (p^k_{ij})^{1-\alpha}} \right]^{\frac{1}{1-\alpha}}. \tag{26}$$

If $\lambda^{\frac{1}{\alpha-1}}$ is substituted in (22) final equation for p^*_{im} is as:

$$p^*_{im} = \frac{1}{\sum_{j=1}^{n} \left(\sum_{k=1}^{L} (p^k_{ij})^{1-\alpha} \right)^{\frac{1}{1-\alpha}}} \left(\sum_{k=1}^{L} (p^k_{im})^{1-\alpha} \right)^{\frac{1}{1-\alpha}}. \tag{27}$$

To simplify the above formula, term $\sum_{j=1}^{n} (\sum_{k=1}^{L} (p^k_{ij})^{1-\alpha})^{\frac{1}{1-\alpha}}$ is represented by r. Therefore:

$$p^*_{im} = \frac{1}{r} \left(\sum_{k=1}^{L} (p^k_{im})^{1-\alpha} \right)^{\frac{1}{1-\alpha}}, \tag{28}$$

where r is a normalization factor, makes summation of each row equal to one. By replacing the term $1-\alpha$ with β, the general equation is as follows:

$$p^*_{im} = \frac{1}{r} \left(\sum_{k=1}^{L} (p^k_{im})^{\beta} \right)^{\frac{1}{\beta}}. \tag{29}$$

TABLE I
DIFFERENT METHODS OF AGGREGATING DISTRIBUTION MATRICES BASED ON RÉNYI DIVERGENCE MEASURE

β	p^*_{im}	*Name*
$\beta \to -\infty$	$\min_k p^k_{im}$	*Minimum*
$\beta = -1$	$\frac{1}{r} \left(\sum_{k=1}^{L} \frac{1}{p^k_{im}} \right)^{-1}$	*Harmonic mean*
$\beta = 0$	$\frac{1}{r} \prod_{k=1}^{L} p^k_{im}$	*Geometric mean(Product)*
$\beta = 1$	$\frac{1}{r} \sum_{k=1}^{L} (p^k_{im})$	*Arithmetic mean*
$\beta = 2$	$\frac{1}{r} \sqrt{\sum_{k=1}^{L} (p^k_{im})^2}$	*Euclidean length*
$\beta \to \infty$	$\max_k p^k_{im}$	*Maximum*

For specific values of β, the above equation converts to simple operators like Minimum, Maximum, Product, etc. Table I represents some of these special cases. It is notable that the attained equation in a different manner can be considered as a combination of probability distributions. In this area, in 1986, Genest and Zidak [31] and later in 1995, Jacobs [32] provided an overview on the aggregation of expert opinions in the case they were considered as probability distributions. Moreover, in [33] Amari introduced the α-mixture of a number of probability distributions $p_i(s), i = 1, 2, \ldots, k$ by

$$\tilde{p}_\alpha(s) = c f_\alpha^{-1} \frac{1}{k} \sum f_\alpha[p_i(s)], \tag{30}$$

where the α representation of probability density function $p(s)$ is given by

$$f_\alpha[p(s)] = \begin{cases} \frac{2}{1-\alpha} p(s)^{(1-\alpha)/2}, & \alpha \neq 1 \\ \log p(s), & \alpha = 1 \end{cases}, \tag{31}$$

and r is a normalization factor. It can be inferred that even though our method and α-mixture are obtained in two completely different ways (the former is based on distance minimization by means of Rényi divergence measure), for $\alpha \neq 1$ by setting the α parameter here to $2\beta + 1$ this mixture degrades to our proposed method. In addition, for $\alpha = 1$ the mixture calculates a summation of probability density functions which is the same as multiplication we are doing in the proposed method for $\beta = 0$ since it adds logarithms of these functions. Besides, using Rényi divergence measure in our method has the advantage of computational simplicity and efficiency. Similarly, Murata and Fujimoto [34] proposed the u-mixture of probability distributions which is derived from Bregman divergence.

B. Proposed Ensemble Method

In the previous section, finding the distribution having the minimum distance from given basic distributions by Rényi divergence measure led to (29). Therefore, by applying this equation as the combination operator in ensemble learning, the final model is expected to be a good representative of basic HMMs.

Thus, for this ensemble learning, first each sequence is trained with Baum-Welch algorithm to obtain $\lambda^{(k)} = (A^{(k)}, B^{(k)}, \pi^{(k)})$, and afterwards, these models are aggregated

by (29). Assume that $a_{ij}^{(k)}$ is the (i,j)th entry of transition matrix for the kth basic model and there are K basic HMMs. The (i,j)th entry of transition matrix in ensemble HMM is obtained by:

$$a_{ij}^{*} = \frac{1}{r_{a_i}} \left(\sum_{k=1}^{K} \left(a_{ij}^{(k)} \right)^{\beta} \right)^{\frac{1}{\beta}} \tag{32}$$

and similarly by:

$$b_{ij}^{*} = \frac{1}{r_{b_i}} \left(\sum_{k=1}^{K} \left(b_{ij}^{(k)} \right)^{\beta} \right)^{\frac{1}{\beta}} \tag{33}$$

and

$$\pi_{i}^{*} = \frac{1}{r_{\pi}} \left(\sum_{k=1}^{K} \left(\pi_{i}^{(k)} \right)^{\beta} \right)^{\frac{1}{\beta}} \tag{34}$$

for the emission and prior matrices, where r_{a_i}, r_{b_i} and r_{π} are normalization factors of ith row for each matrix.

IV. Experimental Results

A. Synthetic Datasets

To examine the proposed HMM ensemble method, the HMM Matlab Toolbox written by Kevin Murphy [35] was used. The methods were compared over a range of parameters. For each parameter set, seed models were generated randomly and these models were used for generating training and test sets. Since these models generate the observation sequences, afterwards they are denoted as True models. The considered parameters in each data set are as follows:

- M, the number of states.
- N, the number of observation symbols.
- T, the length of each observation sequence.
- *trial*, the total number of sample sequences generated from each model.
- *diff*, the number of different samples among *trial* samples in training set. Thus remaining *trial-diff* samples are duplicate.
- *Models*, the number of True models used for sample generation. So each training and test set consists of *trial*$\times$ *Models* sample.
- *SeqToModel*, the number of sequence observations used for training basic HMMs.

The assigned values for each parameter is shown in Table II.

TABLE II
Different Values of Each Parameter Used in Experiments With Synthetic Datasets

Parameter	*Value*
M	$\{4, 8, 12, 16\}$
N	$\{4, 8, 12\}$
T	$\{2, 7, 12, 17\}$
trial	$\{4, 8, 12, 16, 20, 24\}$
diff	$\{1, 4, 8, ..., trial\}$
Models	$\{1, 5, 9, 13, 17\}$
SeqToModel	$\{1, 2, 4\}$

TABLE III
Different Values of Each Parameter Used in Experiments With Real-World Datasets

		Value for each Dataset	
		50 Words	Sign Language
Parameter	*M*	$\{4, 8, 16\}$	$\{4, 8, 16\}$
	N	$\{32, 64\}$	$\{8, 16\}$
	T	50	30
	trial	size of the training set	10
	diff	$trial$	$trial$
	Models	1	1
	seqToModel	1	1

B. Real-World Datasets

In addition to synthetic datasets, real-world applications were also employed to examine the performance of the proposed method. For this purpose two datasets were chosen. One contains 50 word images of handwritten pages and the other 10 words of Australian Sign Language; both provided from UCR time series data [36]. In 50 words dataset, each word is describes by a number of instances where each instance is represented by 4 features and the length of distances are different. In sign language dataset, each word is captured from 20 different signers and each instance is represented by 8 features (x, y, z position of a hand, orientation of a palm, and the folding degree of 4 fingers). The aim of experiments on these datasets is to fit a model per word that best describes their corresponding instances by training a Hidden Markov Model. These HMMs can further be used in other tasks such as classification or clustering. The assumed parameters for running the experiments in this case were the same as synthetic datasets but with different values. The value set for these two dataset is given in Table III. For T and N these values are assigned according to the length of time series.

C. Results and Discussion

The performance of the proposed method was evaluated in comparison with Davis's Ensemble Learning method and the corresponding True model of synthetic datasets for each parameter set by 2-fold cross-validation.

As a measure of accuracy, log-likelihood of the test set was used which describes the logarithm of probability that the given model produces the given sequence(s) calculated by Forward-Backward algorithm [1]:

$$logP(O|\lambda) = log \prod_{k=1}^{K} P(O^{(k)}|\lambda) = \sum_{k=1}^{K} logP(O^{(k)}|\lambda) \tag{35}$$

where $O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ is the set of observation sequences and λ is the given model.

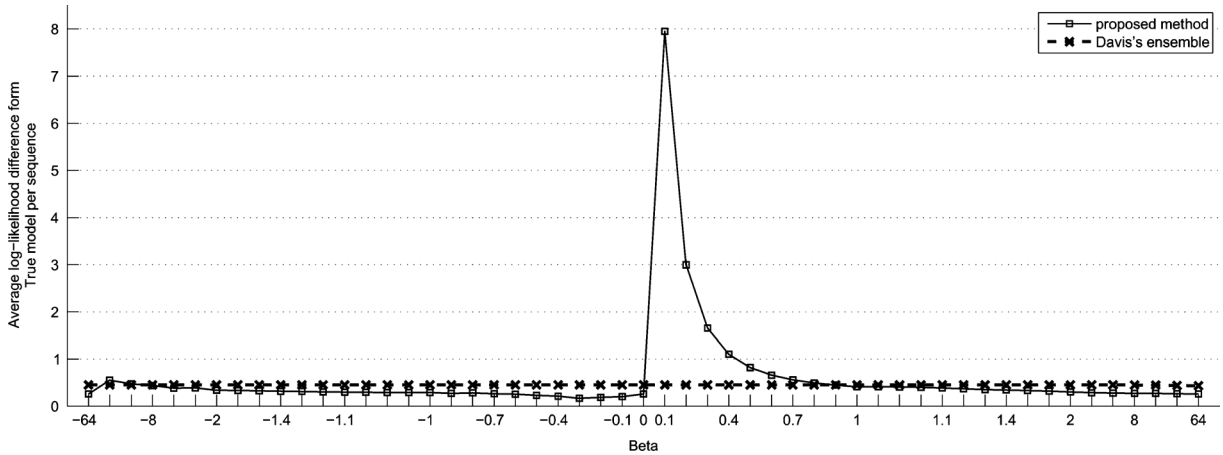

Fig. 1. Log-likelihood differences from True model for Davis's ensemble and proposed method as the values of β varies.

For the proposed method a range of β values was examined and the whole process was repeated 10 times for each parameter set and the average result was considered for evaluation. The set of used values for β is:

$$\begin{aligned}\beta = \{&-64, -32, -16, -8, -4, -3, -2, -1.7, -1.5,\\ &-1.4, -1.3, -1.2, -1.1, -1.05, -1.01, -1.001, -1,\\ &-0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2,\\ &-0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1,\\ &1.001, 1.01, 1.05, 1.1, 1.2, 1.3, 1.4, 1.5,\\ &1.7, 2, 3, 4, 8, 16, 32, 64\}\end{aligned} \tag{36}$$

The reason of this kind of assignment for β is that the critical points for β in (29) are in [−2,2]. As the value goes to $\pm\infty$ the behavior of the β converges, so this interval has been investigated more specifically.

As described before, setting $SeqToModel = \mathrm{k}$ means each basic HMM in our proposed method is obtained by training on k sequence, while in Davis's ensemble according to [18] basic HMMs are always produced by single sequences.

To begin with the analysis of the method for synthetic datasets, the average log-likelihood derived from each parameter set considering $SeqToModel = 1$ is compared with True model per sequence, which means the log-likelihood is divided by the sequence length of its corresponding test set. Fig. 1 illustrates this comparison for two ensemble methods, Davis's method, and the proposed ensemble learning. The values show the log-likelihood difference from True model so, the better the ensemble model is generated, the closer it is to X-axis. The trend of the proposed method for negative β values starts with the close level to Davis's ensemble. As β value gets closer to zero, the proposed method shows better performance by approaching to the True model. Afterwards, with a dramatic increase at $\beta = 0.1$, the trend starts to decrease followed by getting the same value of Davis's ensemble at $\beta = 1$, as is expected based on Table II, and slight decrease with higher values of β. Hence in general, $\beta = -0.3$ has the best performance or closest value to True for the whole range of β. Considering only the positive values, as the β increases it shows better performance.

Next, the proposed method was compared further with Davis's ensemble. The measure used here is the percentage of the cases where the proposed method outperforms Davis's ensemble by comparing their log-likelihood. Evidently, for $\beta = 1$, the proposed model is the same as Davis's ensemble so the percentage of improvement is zero because in all the cases they return equal log-likelihood. Besides this, higher percentage means more improvement. In Fig. 2 this measurement is represented with averaging the results over the whole runs and setting the value of parameter *SeqToModel* to 1. Surprisingly, the highest percentage of improvement is for $\beta = 1.001$, and as the value of β increases, this improvement decreases. Eventually, except the interval [0.1,1], the proposed method outperforms Davis's ensemble.

In conclusion, if the number of improvements is taken as the criterion, the best β achieving the purpose is 1.001 but if the log-likelihood difference is the measure of performance, which is considered here, both $\beta = -0.3$ and $\beta = +\infty$ are suitable.

In continuation of former analysis, the effect of parameters on performance was investigated. Based on above mentioned statement, there were two eligible values for β where $+\infty$ was chosen for analyzing among them. Fig. 3 illustrates variation of performance on each parameter.

To begin with parameter analysis, the number of states were considered. According to Fig. 3(a), as the number of states increases, both the proposed method and Davis's ensemble get closer to the True model while the distance between average log-likelihood of proposed method and Davis's ensemble rises meaning that the proposed method approaches the True model in higher proportion.

Secondly, the number of observation symbols were considered in Fig. 3(b). Even though the performance of both methods decreases with increment of N, the distance between average log-likelihood of proposed method and Davis's ensemble still increases same as Fig. 3(a).

Afterwards, the effect of sequence length and number of models were taken into consideration as is illustrated in Fig. 3(c) and (d), respectively. These parameters have reverse trend of performance as their values increase, but for both of them,

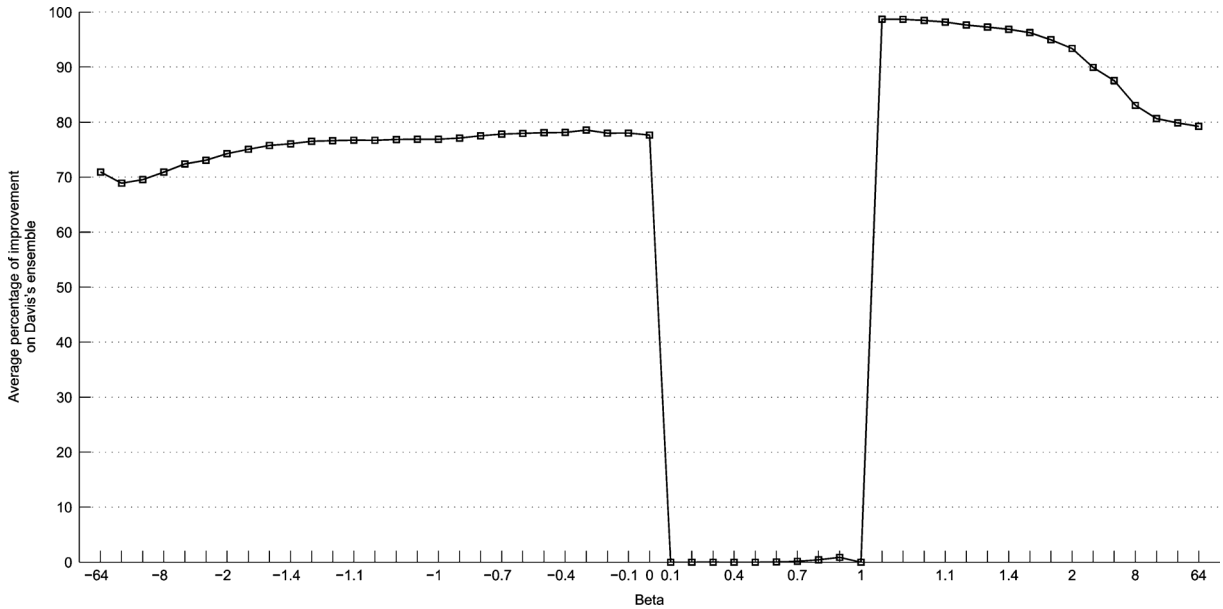

Fig. 2. Comparison between proposed method and Davis's ensemble based on the number of improvements for different values of β.

the distance between average log-likelihood of considered models decreases while their value increase. More precisely, for sequence length the best performance was achieved in its minimum assumed value, and for model numbers the best case of the proposed method and Davis's ensemble, especially the former, was the maximum assigned value. However, even after decrease of performances the proposed method still outperforms the other method and seems to keep its distance from Davis's method. Moreover, the proposed method is able to have the same performance as the Davis's ensemble but with fewer observation symbols and sequence length which means the proposed method needs simpler HMM models on a specific time series dataset to have equal performance with Davis's ensemble method.

The last investigated parameter, *SeqToModel*, is shown in Fig. 4. This parameter was also tried on Davis's ensemble method even though in the main paper [18] it was not included. In short, both methods work better when each basic HMM is produced from a single sequence.

As the last analysis of the synthetic datasets, determining ranges in which the proposed method works better than Davis's ensemble, was investigated for some parameters. To perform such an analysis the comparison measure is defined as follows:

$$Z = \frac{loglike(method) - loglike(Davis)}{|loglike(Davis)|}. \qquad (37)$$

This is the difference between log-likelihood of two methods, divided by log-likelihood of Davis's method, in order to scale the results to improve the comparison. The results of this comparison can be illustrated in the form of contour plots which are shown in Fig. 5 for some parameters. For each analysis, two parameters were considered and the value of Z is averaged for values of all other parameters. In addition, due to definition of Z, the higher the value of comparison measure, the better the performance of the proposed method in comparison with Davis's ensemble. Hence, the lighter areas in contour plots show a greater difference between two methods. In the following, each analysis will be described shortly. Firstly, the number of observation symbols (N) and the number of states (M) were considered as the parameters of analysis. According to Fig. 5(a), for small number of states M, the comparison measure is not very high, meaning the two method do not differ very much (still the proposed method is better). But for a constant N, as the number of states increase, the proposed method shows superior performance. Thus, in general, we can conclude that the proposed method intensely outperforms Davis's ensemble for large Ms and small Ns.

Secondly, the comparison measure was investigated as a function of sequence length (T) and the number of states (M). As in Fig. 5(b), the proposed method works significantly better than Davis's ensemble for small sequence lengths. Nevertheless, it is still the superior method for all ranges of T (from the plot, the measure is always greater than zero).

Thereafter, the two methods were compared considering M and *diff* as the parameters. Fig. 5(c) represents the result of this analysis which shows better performance of the proposed method for smaller value of *diff*. Although the difference is significant for small number of differences in sample sequences, the proposed method outperforms Davis's ensemble for all values of *diff*.

And lastly, the effect of number of sample sequences on comparison measure was explored in Fig. 5(d). shows the general trend as with increasing the number of samples the proposed method's performance increases except for the case where both the number of states, M, and number of samples are small.

In the second step of analysis, the performance of the proposed method on real-world applications was taken into consideration. As mentioned before, the two chosen datasets included images of 50 handwritten words and sign languages of 10 words. The proposed method on these datasets were compared against Davis's Ensemble. Since a better trained model ought to have a better log-likelihood on the unseen given datasets, the attained

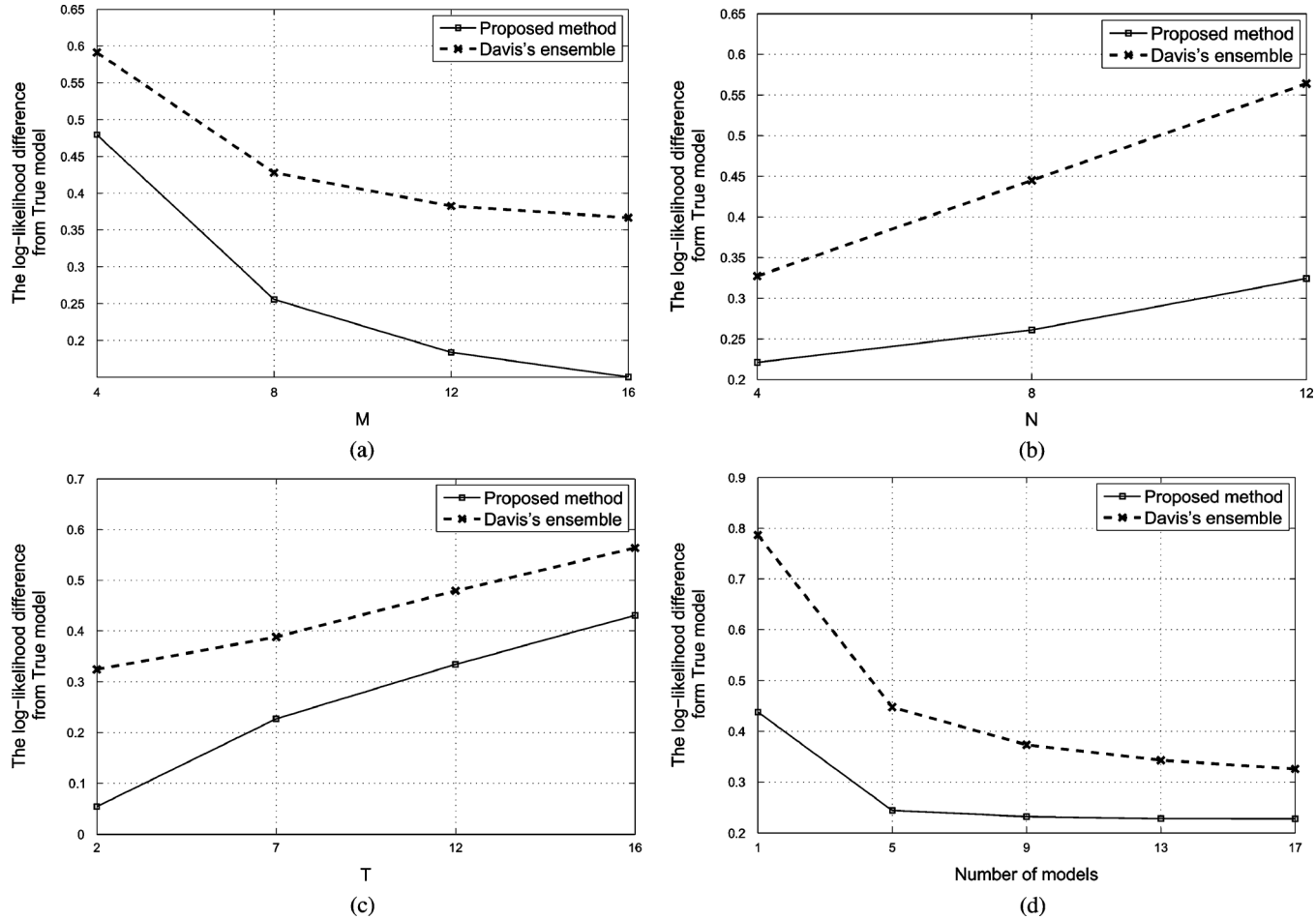

Fig. 3. Changes of performance as the parameters vary. (a) Effect of number of states, (b) Effect of number of observation symbols, (c) Effect of sequence length, (d) Effect of number of models.

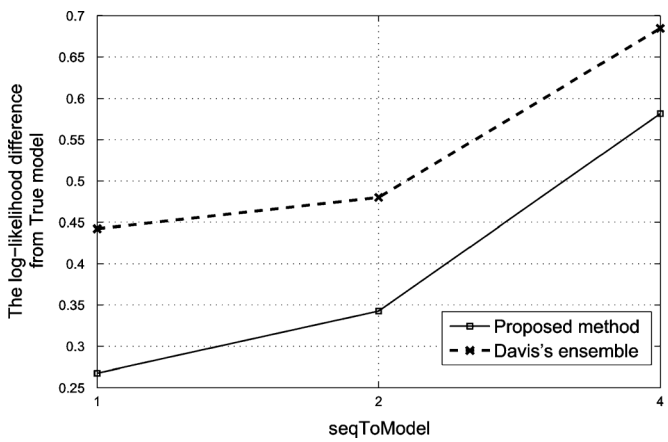

Fig. 4. Difference of log-likelihood with parameter *SeqToModel* variation.

log-likelihoods of these two compared methods on the test set were used as the criterion. Figs. 6, 7 represent the attained results for these two datasets respectively. Considering 50 words dataset, the trend of the proposed method starting with the small values of log-likelihood for negative values of β, increases as it goes further such that it crosses Davis's ensemble gradually and has it highest value for $\beta = -0.3$. Following that, again it starts to decrease slowly as its performance would get worse than Davis's ensemble if the trend continues to infinite values of β but with close distance. Next, for sign language dataset the experiments revealed that the proposed method's trend for the assumed range of β values is almost close to Davis's ensemble except for two points. One is for $\beta = -0.3$ where the proposed method outperforms Davis's and the other is for $\beta = 0.1$ where the proposed method is dramatically worse than Davis's ensemble. So for both datasets the best point where the proposed method outperforms the Davis's ensemble was $\beta = -0.3$ similar to synthetic datasets results. However, it is recommended for each application its best β value be calculated by experiments like cross validation the same as what is done in this paper.

By and large, regarding the goal of ensemble learning, we can therefore claim that our proposed method is a better ensemble in comparison with Davis's ensemble and besides this the proposed method meet the performance of Davis's method with simpler HMM models which also means time efficiency as a result.

V. Conclusion and Future Work

To conclude, a new ensemble learning method is presented in this paper. The proposed method looks at every row of the HMM parameters as a probability distribution and tries to find a distribution with minimum distance to all the given training

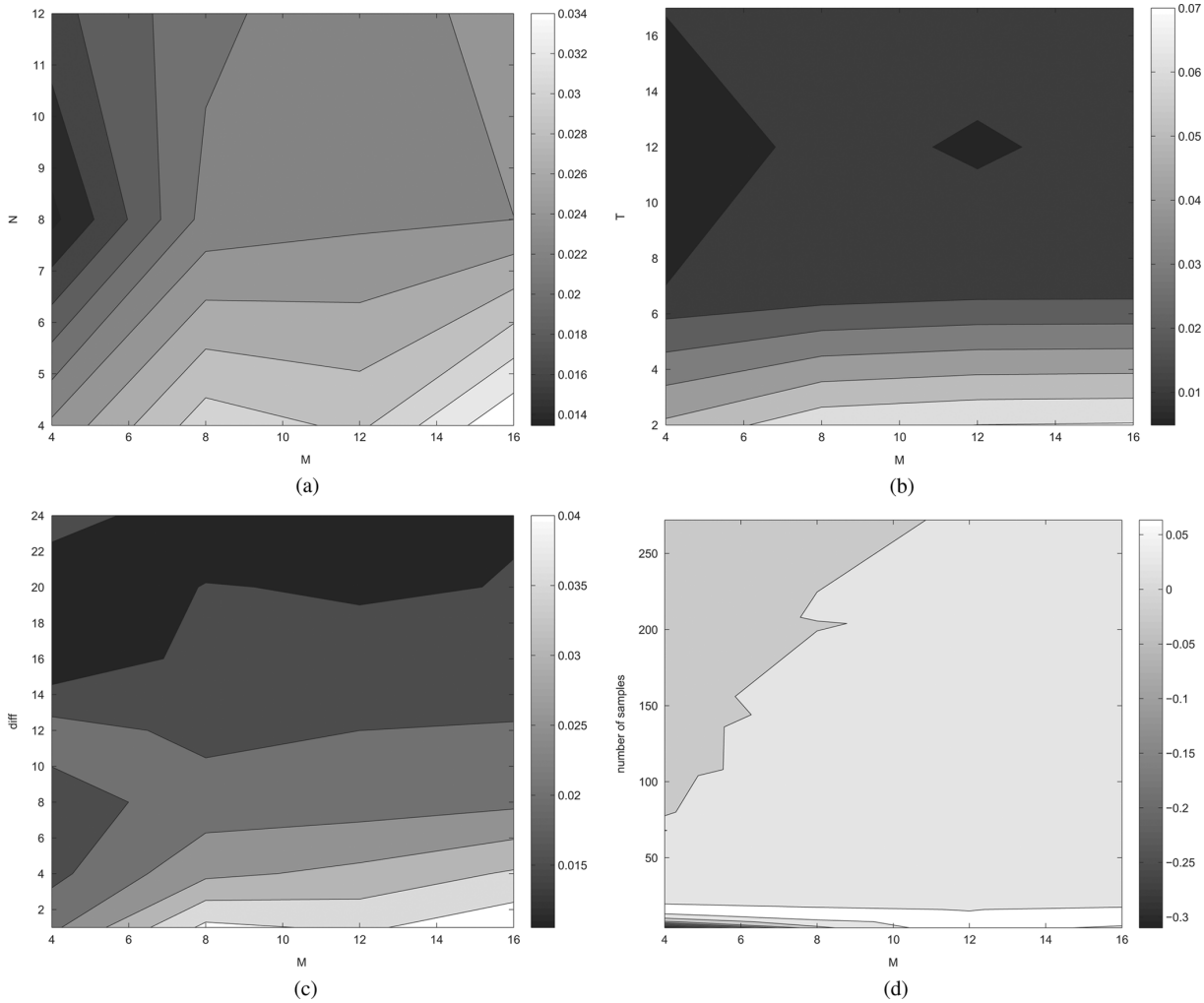

Fig. 5. Comparing the proposed method and Davis's ensemble in order to find out some rules about good ranges via comparison measure defined in (37). (a) Comparison measure as a function of N and M, by averaging the measure for other parameters. (b) Comparison measure as a function of T and M, by averaging the measure for other parameters. (c) Comparison measure as a function of *diff* and M, by averaging the measure for other parameters. (d) Comparison measure as a function of number of samples $(trial \times Models)$ and M, by averaging the measure for other parameters.

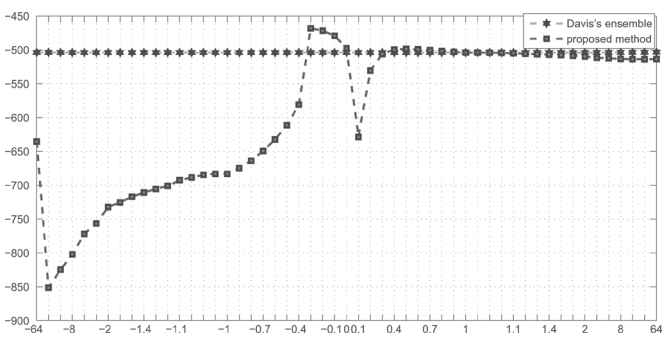

Fig. 6. Comparison between log-likelihood of compared methods with 50 words dataset for different values of β.

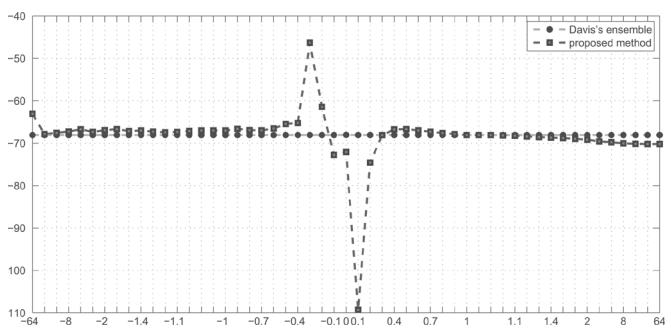

Fig. 7. Comparison between log-likelihood of compared methods with sign language dataset for different values of β.

ones by means of Rényi Divergence Measure. To evaluate the performance of the method, several comparisons with Davis's ensemble which is a simple ensemble-average learning, were made. The results, as discussed in the related section, confirm the efficiency of the proposed method for different values of the parameters. In short, the superiority of the proposed method to ensemble-average learning is more significant when the number of states and observation symbols increase and when the number of generating models, sequence model, and sequence length is low. Moreover, the overall experiments revealed that the best value for parameter β is $\beta = -0.3$. However, for some applications there might be other better values that should be attained by doing validate experiments on reasonable training sets. In future, we will explore the efficiency of this proposed learning

method on the task of time series supervised and unsupervised classification along with application on real-world classification problems. Furthermore, instead of Rényi divergence measure other divergence measures such as f-divergence and its instances or k-divergence can be examined to investigate possible improvements.

REFERENCES

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

[2] A. Krogh, M. Brown, I. S. Mian, and K. Sjoelander, "Hidden Markov models in computational biology: Applications to protein modeling," *Molecular Biol.*, vol. 235, no. 5, p. 1501, 1994.

[3] S. R. Eddy, "Multiple alignment using hidden Markov models," in *Proc. Int. Conf. Intell. Syst. Molecular Biol.*, 1995, vol. 3, pp. 114–120.

[4] P. Frasconi, G. Soda, and A. Vullo, "Hidden Markov models for text categorization in multi-page documents," *Intell. Inf. Syst.*, vol. 18, pp. 195–218, 2002.

[5] A. Wilson and A. Bobick, "Hidden Markov models for modeling and recognizing gesture under variation," *Pattern Recognit. Artif. Intell. (IJPRAI)*, vol. 15, no. 1, pp. 123–160, Feb. 2001.

[6] A. Nefian and M. Hayes, "Hidden Markov models for face recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process. ICASSP98*, Seattle, WA, USA, 1998, pp. 2721–2724.

[7] J. J. Lee, J. Kim, and J. H. Kim, "Data-driven design of hmm topology for online handwriting recognition," *Pattern Recognit. Artif. Intell.*, vol. 15, no. 1, pp. 107–121, 2001.

[8] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden Markov model," *Proc. IEEE Signal Process.*, vol. 48, no. 2, pp. 517–533, 2000.

[9] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annal. Math. Statist.*, vol. 37, pp. 1554–1563, 1966.

[10] L. Baum and J. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol Soc.*, vol. 73, pp. 360–363, 1967.

[11] L. Baum and G. Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[12] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[13] L. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1970.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[15] S. Levinson, L. Rabiner, and M. Sondhi, "An introduction to the application of the theory of probabilistic functions of Markov process to automatic speech recognition," *Bell Syst. Techn. J.*, vol. 63, no. 4, pp. 1035–1074, 1983.

[16] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.

[17] D. J. C. Mackay, Ensemble Learning for Hidden Markov Models Cavendish Lab., Univ. Cambridge, U.K., Tech. Rep., 1997.

[18] R. I. A. Davis, B. C. Lovell, and T. Caelli, "Improved estimation of hidden Markov model parameters from multiple observation sequences," in *Proc. Int. Conf. Pattern Recognit. (ICPR2002)*, Quebec City, QC, Canada, Aug. 2002, pp. 168–171.

[19] X. D. Hoang and J. Hu, "An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls," in *Proc. IEEE Int. Conf. Netw.*, 2004, pp. 470–474.

[20] X. D. Hoang, J. Hu, and P. Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," *Netw. Comput. Appl.*, vol. 32, no. 6, pp. 1219–1228, 2009.

[21] R. I. Davis and B. C. Lovel, "Comparing and evaluating hmm ensemble training algorithms using training and test and condition number criteria," *Pattern Anal. Appl.*, vol. 6, no. 4, pp. 327–336, 2003.

[22] R. I. A. Davis and B. C. Lovell, "Improved ensemble training for hidden Markov models using random relative node permutations," in *Proc. Workshop Dig. Image Comput. (WDIC2003)*, Brisbane, Australia, Feb. 2003, pp. 83–86.

[23] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[24] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[25] Y. Freund, "Boosting a weak learning algorithm by majority," in *Proc. 3rd Annu. Workshop Computa. Learn. Theory*, 1995.

[26] X. Jian, W. Jian-Guang, Z. Yue-Ting, and W. Fei, "Ensemble learning hmm for motion recognition and retrieval by isomap dimension reduction," *Zhejiang University SCIENCE A*, vol. 7, no. 12, pp. 2063–2072, 2006.

[27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. 2th Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.

[28] R. I. A. Davis, C. J. Walder, and B. C. Lovell, "Improved classification using hidden Markov averaging from multiple observation sequences," in *Proc. Workshop Signal Process. Appl. (WOSPA2002)*, Brisbane, Australia, 2002.

[29] A. Rényi, "On measures of information and entropy," in *Proc. 4th Berkeley Symp. Math., Statist. Probability*, 1960, pp. 547–561.

[30] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[31] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and an annotated bibliography," *Statist. Sci.*, vol. 1, no. 1, pp. 114–135, 1986.

[32] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Comput.*, vol. 7, no. 5, pp. 867–888, 1995.

[33] S.-I. Amari, "Integration of stochastic models by minimizing α-divergence," *Neural Comput.*, vol. 19, no. 10, pp. 2780–2796, 2007.

[34] N. Murata and Y. Fujimoto, Bregman Divergence and Density Integration p. 2009B-3, 2009.

[35] K. Murphy, Hidden Markov Model (HMM) Toolbox for Matlab 1998 [Online]. Available: http://www.cs.ubc.ca/murphyk/Software/HMM/hmm.html

[36] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana, The UCR Time Series Classification/Clustering 2011 [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_Series_data

Nazanin Asadi received the B.Sc. and M.Sc. degrees in computer engineering from Isfahan University of Technology, Isfahan, Iran in 2010 and 2012 respectively. Her research interests include Pattern Recognition and Machine Learning.

Abdolreza Mirzaei received the B.Sc., M.Sc. and Ph.D. degrees in 2001, 2003, and 2009, respectively, from Isfahan University, Iran University of Science and Technology, and Amirkabir University of Technology. He is currently an assistant professor in the electrical and computer engineering department at Isfahan University of Technology, Iran. His research areas include Pattern Recognition, Machine Learning, Data Mining, and Swarm Intelligence.

Ehsan Haghshenas received the B.Sc. degree in computer engineering (major in Software) from Isfahan University of Technology, Iran in 2012. He was with Isfahan Mathematics House, Iran as a teacher from 2009 to 2012. He is currently a M.Sc. candidate at the Computer Science department, University of Western Ontario, Canada. His research interests include Machine Learning and its applications, and Computational Biology (Bioinformatics).