Signal Processing 94 (2014) 490–497

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A novel and robust parameter training approach for HMMs under noisy and partial access to states

Huseyin Ozkan [a,b,*], Arda Akman [c,1], Suleyman S. Kozat [a]

[a] Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey
[b] Department of Image Processing at MGEO Division, Aselsan Inc., Ankara, Turkey
[c] Turk Telekom Group R&D, Ankara, Turkey

ARTICLE INFO

Article history:
Received 27 September 2012
Received in revised form
7 July 2013
Accepted 12 July 2013
Available online 24 July 2013

Keywords:
HMM
ML estimator
Incomplete data
Partially observed states

ABSTRACT

This paper proposes a new estimation algorithm for the parameters of an HMM as to best account for the observed data. In this model, in addition to the observation sequence, we have *partial* and *noisy* access to the hidden state sequence as side information. This access can be seen as “partial labeling” of the hidden states. Furthermore, we model possible mislabeling in the side information in a joint framework and derive the corresponding EM updates accordingly. In our simulations, we observe that using this side information, we considerably improve the state recognition performance, up to 70%, with respect to the “achievable margin” defined by the baseline algorithms. Moreover, our algorithm is shown to be robust to the training conditions.

1. Introduction

In a wide variety of applications in time series analysis ranging from speech processing [1–8], bioinformatics [9,10] to natural language processing [11–14], the observation sequence is represented as a stochastic process, depending on another stochastic process that generates a sequence of hidden (unobserved) states. With certain conditional independence properties regarding the observations as well as the states, this is known as Hidden Markov Model (HMM) [1,15]. In this paper, we particularly concentrate on discrete-time finite-state HMM with finite alphabet, which is described by two families of random variables: the hidden state sequence Z_t and the observation sequence Y_t. The random variables in the state sequence Z_t form a stochastic, discrete-time Markov chain and the observation Y_t, conditioned on the present hidden state Z_t, is independent with the past and future observations as well as the hidden states. The corresponding conditional independence structure of the model is shown as a directed acyclic graph [16] in Fig. 1a. Hence, an HMM is completely characterized by the set of parameters $\lambda=(\pi,A,B)$, where A_{ij} is the state transition probabilities, B_{ij} is the observation emission probabilities and π_i is the initial state probabilities. A detailed description of the model can be found in [1]. Estimation of these model parameters $\lambda=(\pi,A,B)$ is an important problem in applications using HMM [1,6,7,9–14,17,18]. Since there is no closed form solution for the set of parameters that maximizes the probability of the observation sequence given the model, instead, iterative algorithms such as the Expectation-Maximization (EM) algorithm[19,20] (or equivalently the Baum–Welch method [21]) is used to obtain a local optimal solution [1]. In this paper, we derive a new set of iterative EM equations that yield a locally optimal solution for the model parameters, when the ordinary model of the observation sequence, e.g., as in [1], is different. In our model, in addition to the observation sequence y_t (upper case letters are used to denote the random variables and the lower case letters are used

* Corresponding author at: Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey. Tel.: +90 312 290 1219, +90 312 847 5300..

E-mail addresses: hozkanafl@gmail.com, huseyin@ee.bilkent.edu.tr, huozkan@aselsan.com.tr (H. Ozkan), arda.akman@turktelekom.com.tr (A. Akman), kozat@ee.bilkent.edu.tr (S.S. Kozat).

[1] Tel.: +90 312 555 6700.

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.07.015

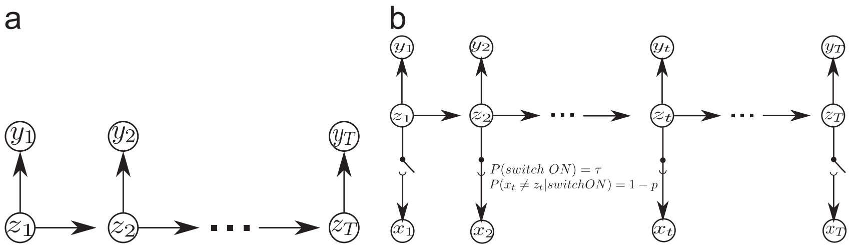

Fig. 1. (a) The conditional independence structure of an HMM with discrete-time finite-states z_t and observations y_t of a finite alphabet. (b) The conditional independence structure of an HMM with partial and noisy access x_t to the state sequence.

to denote the corresponding realizations), we observe a part of the hidden state sequence as side information. More precisely, at every time instant t, we observe the hidden state z_t as x_t with probability τ, i.e., with $1-\tau$ probability the state stays hidden. This gives partial access to the state sequence as side information which is, in our work, incorporated in the corresponding parameter estimation problem and the associated EM algorithm. We emphasize that the state observations are not necessarily confined to a time interval but may even be sparsely and randomly distributed along the complete time span of the application. In the limiting case, if τ is 0, then there would be no state observation, and we recover the ordinary, unsupervised HMM training. Therefore, our model provides a generalized framework by letting partial access to the state sequence. Moreover, we also allow that a state observation might be corrupted with noise such that if z_t is ever observed then $P(X_t \neq z_t | Z_t = z_t) = 1-p$. Then the corresponding conditional independence structure in this case of partially observable states is shown in Fig. 1b. Under these new circumstances, we explicitly provide the mathematical derivations of the new set of iterative EM equations that incorporates the side information and estimate the model parameters accordingly. In these derivations, the probability that a state observation is incorrect, $1-p$, is assumed to be known and it is provided to our algorithm as a parameter, p, which defines the confidence on the side information. Simulations show that our method is robust to the confidence parameter p, even if it does not exactly match with the underlying true quality of the side information, p_{true}.

Since the hidden state sequence is partially observed, our work falls into the category of Partially Hidden Markov Model (PHMM) training (note that this term is used in [22] in a different context). Similar to semi-supervised learning, PHMMs use both "labeled" (in our context the state information) and "unlabeled" data to obtain improved model training. Such an approach is suitable, when we have access to a limited amount of labeled data along with a large amount of unlabeled data. This happens, as an example, in speech processing applications [8], where labeling, i.e., transcription, is naturally costly [8,23], hence only limited amount is affordable, and transcriptions may contain errors. Furthermore, by allowing noisy access to the states, we model "mislabeling" event that may occur during labeling stage. PHMMs, to the best of our knowledge, date back to the studies [12–14] in the area of Natural Language Processing. In these studies, tagged text, corresponding to the known states of a PHMM, is first analyzed through a relative frequency modeling to construct an initial model, then this model is fed into the ordinary HMM training algorithm. However, these studies do not rigorously show how the partial state information is incorporated within the ordinary HMM parameter estimation framework. The Maximum Likelihood Estimator (MLE) for the model parameters in a special case of PHMMs, where only a certain state from the state space in the underlying Markov chain is known, is theoretically (consistency and asymptotic normality of the estimator) analyzed in [11]. However, the equations for computing the MLE (using the EM algorithm or other Likelihood maximization techniques) in this special case of PHMM are not derived. In [18], iterative EM equations for a general case, where each observation can only belong to a pre-defined set of acceptable states are given, but no complete derivation is provided. On the contrary, we explicitly derive the new set of iterative EM equations for the PHMM parameter estimation problem, when there is partial access to the underlying hidden state sequence. Furthermore, the partial observation of the state sequence might be prune to noise in our model and this case is not considered in the existing literature.

After we provide the brief description of the basic HMM framework and the parameter estimation equations in Section 2, we derive the new set of iterative EM equations that incorporates partial and noisy access to the state sequence as side information in Section 3. Simulations are presented in Section 4 and the paper concludes with final remarks in Section 5.

2. Problem description

In this section, we briefly describe the basic framework for the HMM parameter estimation problem [1]. For the sake of notational simplicity, we study discrete-time finite-state HMM with finite alphabet. However, our derivations for incorporating the side information in Section 3 can be readily extended to the case, where the observations come from a continuous distribution and outcomes are vectors. A discrete-time HMM with finite alphabet is formally a Markov model, for which we have a sequence of observations, y_t, drawn from a finite alphabet $V = \{v_1, v_2, \ldots, v_{N_v}\}$, i.e., $y_t \in V, 1 \leq t \leq T$. We also have a sequence of hidden (unobserved) states $z_t \in S = \{s_1, s_2, \ldots, s_{N_s}\}$, where S is the set of possible states, generated from a Markov process. Namely, $P(Z_t = z_t | \mathbf{Z}_1^{t-1} = \mathbf{z}_1^{t-1}) = P(Z_t \; = z_t \; | Z_{t-1} = z_{t-1})$, where (and in this paper) the upper case (bold) letters are used to denote (a collection of) random variables and the lower case (bold) letters denote the corresponding (collection of) realizations, i.e., $\mathbf{z}_1^{t-1} = \{z_1, z_2, \ldots, \; z_{t-1}\}$, similarly for $\mathbf{Z}_1^{t-1}$. The observation sequence $y_1, y_2, \dots, y_t$ is generated based on the state sequence $z_1, z_2, \dots, z_t$, i.e., $P(Y_t = y_t | \mathbf{Z}_1^t = \mathbf{z}_1^t, \mathbf{Y}_1^{t-1} = \mathbf{y}_1^{t-1}) = P(Y_t = y_t | Z_t = z_t)$. We consider A as the transition matrix, where A_{ij} represents the transition probability from state s_i to s_j, $A_{ij} = P(Z_t = s_j | Z_{t-1} = s_i)$. Similarly, B is the observation probabilities at each state, i.e., $B_{ij} = P(Y_t = v_j | Z_t = s_i)$. In order to complete the HMM observation model, we also define the initial state probabilities as $\pi_i = P(Z_1 = s_i)$. Thus, $\lambda = (\pi, A, B)$ represents the parameter set that completely characterizes the HMM model as shown in Fig. 1a.

As for the HMM parameter estimation problem, the Maximum Likelihood (ML) estimation, $\arg\max_\lambda P(\mathbf{Y} = \mathbf{y} | \lambda)$, $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$, is locally solved iteratively using the Expectation Maximization (EM) algorithm [1]. Then the iterative re-estimation formulas for the HMM parameters providing the ML estimate are as follows:

$$\hat{A}_{ij} = \frac{\sum_{t=1}^{T-1} \epsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad \hat{B}_{ij} = \frac{\sum_{t=1}^{T-1} 1_{\{y_t = v_j\}} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad \hat{\pi}_i = \gamma_1(i). \tag{1}$$

Here, $\epsilon_t(i,j)$ is defined as the probability of transition at time t from state s_i to s_j, given the observations $\mathbf{y}$ and the current parameters of the model, i.e.,

$$\epsilon_t(i,j) = P(Z_t = s_i, Z_{t+1} = s_j | \mathbf{Y} = \mathbf{y}, \lambda),$$

and $\gamma_t(i)$ is defined as the probability of being at state $z_t{=}s_i$, given the observations and the model, i.e.,

$$\gamma_t(i) = \sum_{j=1}^{N_s} \epsilon_t(i,j).$$

These iterative re-estimation formulas can be computed efficiently through the well-known forward–backward procedure [24,25], which is based on the forward and backward variables and the corresponding recursions. The forward variable, $\alpha_t(i)$, along with the recursion in [1] is given by the following:

$$\alpha_t(i) = P(\mathbf{Y}_1^t = \mathbf{y}_1^t, Z_t = s_i | \lambda) = B_{iy_t} \sum_{j=1}^{N_s} \alpha_{t-1}(j) A_{ji},$$

$$\alpha_1(i) = \pi_i B_{iy_1}, \quad 2 \le t \le T,$$

which is the probability of observing $\mathbf{y}_1^t$ and being at state $z_t{=}s_i$, given the model λ. Similarly, the backward variable is given by the following:

$$\beta_t(i) = P(\mathbf{Y}_{t+1}^T = \mathbf{y}_{t+1}^T | Z_t = s_i, \lambda)$$
$$= \sum_{j=1}^{N_s} \beta_{t+1}(j) A_{ij} B_{jy_{t+1}}, \quad \beta_T(i) = 1, \ 1 \le t \le T-1,$$

which is the probability of observing $\mathbf{y}_{t+1}^T$, given the state $z_t{=}s_i$ and the model. By noting that $P(Z_t = s_i, Z_{t+1} = s_j, \mathbf{Y} = \mathbf{y} | \lambda) = \alpha_t(i) A_{ij} B_{jy_{t+1}} \beta_{t+1}(j)$ and $P(\mathbf{Y} = \mathbf{y} | \lambda) = \sum_{k=1}^{N_s} \sum_{l=1}^{N_s} \alpha_t(k) A_{kl} B_{ly_{t+1}} \beta_{t+1}(l)$, we obtain

$$\epsilon_t(i,j) = \frac{\alpha_t(i) A_{ij} B_{jy_{t+1}} \beta_{t+1}(j)}{\sum_{k=1}^{N_s} \sum_{l=1}^{N_s} \alpha_t(k) A_{kl} B_{ly_{t+1}} \beta_{t+1}(l)}.$$

Then the iterative re-estimation formulas for the HMM parameters given in (1) can be computed efficiently using the forward–backward recursions. As a result, given the training data, we estimate the HMM parameters λ by the iterative re-estimation procedure defined by the EM algorithm. Namely, given the HMM parameters λ_{q-1} at an iteration q, we re-estimate the model parameters as λ_q using the re-estimation formulas in (1). This procedure is guaranteed to improve the likelihood of the observations at every iteration and converge to a set of HMM parameters $\hat{\lambda}$, which is at least locally optimal (cf. [1] and the references therein).

In the following section, we incorporate the noisy side information into the HMM framework. To this end, we introduce the "incomplete-data problem" [19], derive the conditional expectation of the complete-data log likelihood to obtain the iterative re-estimation formulas and finally adapt the forward–backward procedure for the case of partially observable states.

3. HMM training with noisy and partial access to the state sequence

In this section, we derive the new set of iterative EM equations for the HMM parameter estimation, when we have noisy and side information on the hidden states. Here, we have an observation sequence $y_t \in \mathbf{y} = \{y_1, y_2, \dots, y_T\}$, with *partial* and *noisy* access to the hidden states, $z_t \in \mathbf{z} = \{z_1, z_2, \dots, z_T\}$, as this side information. Each hidden state $z \in \mathbf{z}$ might be observed as x with probability τ, i.e., we do not necessarily have a state observation at a given time instant. Hence, we have *partial* access to the hidden state sequence. In addition to this partial access, a state observation x might also be *noisy* such that $P(X \ne s | Z = s) = (1-p)$. We assume that if a state observation is erroneous, then $P(X = s_2 | Z = s_1) = 1/(N_s - 1)$, $\forall s_1, s_2 \in S$ and $s_1 \ne s_2$. We here note that if the probability distribution of the erroneous state observations concentrated at a particular state (for each observed hidden state), then we would have more information about the underlying hidden states. For an instance, suppose we observed $x{=}s$ and it is erroneous, i.e., $z \ne s$. Then, z would be more likely to be the state s^* for which the corresponding erroneous state observations concentrated at $s \ne s^*$. This clearly would be a favorable case in terms of the HMM parameter estimation as well as the recognition of the underlying hidden states. In this paper, we targeted the worst case, i.e., the case of most ambiguous state observations, when there is an error. Hence, we assumed that the probability distribution of erroneous state observations is not concentrated and so uniform. For ease of notation, we define the state observations at every time t as $x_t \in \mathbf{x} = \{x_1, x_2, \dots, x_T\}$, such that if z_t is ever observed as $s \in S$, then $x_t = s$. Otherwise, $x_t = s_0$, where s_0 is a pseudo-state. This expands our state space to $S' = S \cup \{s_0\}$. Thus, we model mislabeling and partial labeling jointly in one complete framework as shown in Fig. 1b.

After having described our model, in the following, we first deduce the iterative re-estimation formulas of the HMM parameters under partial and noisy access to the hidden states. In the following, we consider the HMM as an instance of the "incomplete data problem" and derive the conditional expectation of the complete data log-likelihood to apply the EM algorithm to the Maximum Likelihood estimation. Then we present the forward and backward recursions for an efficient computation of the deduced re-estimation formulas.

3.1. The re-estimation formulas for the HMM parameters through likelihood maximization under partial and noisy access to the states

The ML estimation of the HMM parameters in this case of partial and noisy access to the hidden states is given by the maximization of the log-likelihood of all observations with respect to the model parameters

$$\arg\max_{\lambda} \log(P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)).$$

Clearly, this maximization would have been computationally more tractable if the hidden states $\mathbf{z}$ were completely known, i.e., if we had $x_t = z_t$ at each time t, in addition to the observation sequence $\mathbf{y}$, since then the corresponding conditional probability distribution could be factorized into a simpler form (due to Markov property). Hence, considering the unobserved hidden states as the missing data, it is more plausible to formulate this parameter estimation problem as an instance of the "incomplete data problem" [19] and consider the corresponding augmentation of the observations with the states as $\mathbf{w}=(\mathbf{y},\mathbf{x},\mathbf{z})$ as the "complete data". Then one can construct the following relationship between the incomplete-data log-likelihood $\log(P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda))$ and the complete-data log-likelihood $\log(P(\mathbf{W}=\mathbf{w}|\lambda))$:

$$\begin{aligned}\log(P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)) &\geq \sum_{\mathbf{z}} Q(\mathbf{z};\lambda)\log(P(\mathbf{W}=\mathbf{w}|\lambda)) \\ &\quad - \sum_{\mathbf{z}} Q(\mathbf{z};\lambda)\log(Q(\mathbf{z};\lambda)) \\ &= E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda}\{(\log(P(\mathbf{W}=\mathbf{w}|\lambda)))+C(\lambda)\},\end{aligned}$$

where $Q(\mathbf{z};\lambda)=P(\mathbf{Z}=\mathbf{z}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda)$, $C(\lambda)=-\sum_{\mathbf{z}} Q(\mathbf{z};\lambda)\log(Q(\mathbf{z};\lambda))$ and the expectation is with respect to the random variables $\mathbf{Z}$ conditioned on $(\mathbf{Y},\mathbf{X},\lambda)$. Based on this construction, we can readily maximize the conditional expectation of the complete data log-likelihood through the EM algorithm (cf. [19] and the references therein) in order to maximize the incomplete data likelihood as originally intended. The EM algorithm works iteratively between two separate steps, known as E-steps and M-steps, such that at iteration q, the E-step calculates $Q(\mathbf{z};\lambda_{q-1})$ and the M-step maximizes the conditional expectation $E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda_{q-1}}(\log(P(\mathbf{W}=\mathbf{w}|\lambda_q)))$ with respect to the λ_q and note that $C(\lambda_{q-1})$ does not affect the maximization in the M-step of iteration q. We emphasize that since the complete data log-likelihood here is factorizable, the ML estimation of the HMM parameters becomes computationally more tractable when it is posed as an incomplete data problem. Indeed, we show that the forward–backward procedure stays applicable in this case of partial and noisy access to the states. We now deduce the re-estimation formulas for the HMM parameters.

Let $Q(\mathbf{z};\lambda_{q-1})=P(\mathbf{Z}=\mathbf{z}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})$ be the output of E-step, then M-step carries out the following maximization:

$$\begin{aligned}&\arg\max_{\lambda_q} E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda_{q-1}}(\log(P(\mathbf{W}=\mathbf{w}|\lambda_q))) \\ &\quad = \arg\max_{\lambda_q}\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\log(P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\mathbf{Z}=\mathbf{z}|\lambda_q)),\end{aligned}$$

which, using the product of conditional probabilities, yields

$$\begin{aligned}&\arg\max_{\lambda_q} E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda_{q-1}}(\log(P(\mathbf{W}=\mathbf{w}|\lambda_q))) \\ &\quad = \arg\max_{\lambda_q}\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\log(P(\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x},\mathbf{Z}=\mathbf{z},\lambda_q) \\ &\quad P(\mathbf{X}=\mathbf{x}|\mathbf{Z}=\mathbf{z},\lambda_q)P(\mathbf{Z}=\mathbf{z}|\lambda_q)).\end{aligned}$$

Since $\mathbf{X}$ is independent with λ_q conditioned on $\mathbf{Z}$ and $\mathbf{Y}$ is independent with $\mathbf{X}$ conditioned on $(\mathbf{Z},\lambda_q)$, we obtain

$$\begin{aligned}&\arg\max_{\lambda_q} E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda_{q-1}}(\log(P(\mathbf{W}=\mathbf{w}|\lambda_q))) \\ &\quad = \arg\max_{\lambda_q}\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\log(P(\mathbf{Y}=\mathbf{y}|\mathbf{Z}=\mathbf{z},\lambda_q) \\ &\qquad P(\mathbf{X}=\mathbf{x}|\mathbf{Z}=\mathbf{z})P(\mathbf{Z}=\mathbf{z}|\lambda_q)),\end{aligned}$$

where we can drop the factor $P(\mathbf{X}=\mathbf{x}|\mathbf{Z}=\mathbf{z})$ since it does not depend on λ_q and reach

$$\begin{aligned}&\arg\max_{\lambda_q} E_{\mathbf{Z}|\mathbf{Y},\mathbf{X},\lambda_{q-1}}(\log(P(\mathbf{W}=\mathbf{w}|\lambda_q))) \\ &\quad = \arg\max_{\lambda_q}\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\log(P(\mathbf{Y}=\mathbf{y}|\mathbf{Z}=\mathbf{z},\lambda_q)P(\mathbf{Z}=\mathbf{z}|\lambda_q)).\end{aligned} \tag{2}$$

We point out that the maximization in (2) does not involve the side information $\mathbf{x}$, except that $Q(\mathbf{z};\lambda_{q-1})$ is related to $\mathbf{x}$. However, since $Q(\mathbf{z};\lambda_{q-1})$ is calculated in E-step before M-step starts in the course of our algorithm, it only brings constant factors to the maximization in (2) and, hence, it does not affect the M-step derivations. Therefore, rest of the derivations follows the regular M-step derivations of the EM algorithm for the ordinary HMM parameter training and we estimate the transition probabilities as

$$\begin{aligned}\hat{A}_{ij} &= \frac{\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\sum_{t=1}^{T-1} 1_{\{z_t = s_i \wedge z_{t+1} = s_j\}}}{\sum_{\mathbf{z}} Q(\mathbf{z};\lambda_{q-1})\sum_{t=1}^{T-1} 1_{\{z_t = s_i\}}} \\ &= \frac{\sum_{\mathbf{z}}\sum_{t=1}^{T-1} 1_{\{z_t = s_i \wedge z_{t+1} = s_j\}} P(\mathbf{Z}=\mathbf{z}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})}{\sum_{\mathbf{z}}\sum_{t=1}^{T-1} 1_{\{z_t = s_i\}} P(\mathbf{Z}=\mathbf{z}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})},\end{aligned}$$

where $1_{\{h\}}$ is the indicator function such that it outputs 1 if h, as a statement, is satisfied; and 0 otherwise. Here, the indicator function in the numerator and the denominator marginalizes the probability $P(\mathbf{Z}=\mathbf{z}|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})$, since the outer summation is over all possible hidden state sequences. Hence, we obtain

$$\begin{aligned}\hat{A}_{ij} &= \frac{\sum_{t=1}^{T-1} P(Z_t = s_i, Z_{t+1} = s_j|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})}{\sum_{t=1}^{T-1} P(Z_t = s_i|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda_{q-1})} \\ &= \frac{\sum_{t=1}^{T-1}\overline{\epsilon}_t(i,j)}{\sum_{t=1}^{T-1}\overline{\gamma}_t(i)}.\end{aligned}$$

Here, $\overline{\epsilon}_t(i,j)$ is the probability of transition at time t from state s_i to s_j, given the observations $\mathbf{y}$, the side information $\mathbf{x}$, and the model λ_{q-1}, i.e.,

$$\overline{\epsilon}_t(i,j) = P(Z_t = s_i, Z_{t+1} = s_j|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda), \tag{3}$$

and $\overline{\gamma}_t(i)$ is the probability of being at state $z_t = s_i$, given the observations, the side information and the model λ_{q-1}, i.e.,

$$\overline{\gamma}_t(i) = P(Z_t = s_i|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda) = \sum_{j=1}^{N_s}\overline{\epsilon}_t(i,j).$$

Note that the state transition probabilities are estimated, given the side information, as the expected number of transitions from state s_i to s_j divided by the expected number of times in the state s_i. Similarly, $\hat{B}_{ij}$ is given by

$$\hat{B}_{ij}=\frac{\sum_{t=1}^{T-1}1_{\{y_t=v_j\}}\overline{\gamma}_t(i)}{\sum_{t=1}^{T-1}\overline{\gamma}_t(i)},$$

which is, given the side information, the expected number of times in the state s_i and observing v_j, divided by the expected number of times in the state s_i. Also, the set of initial probabilities for the hidden state z_1 is estimated as $\hat{\pi}_i=\overline{\gamma}_1(i)$. We next derive the forward and backward recursions in order to efficiently compute the EM estimate of the HMM parameters.

3.2. Forward and backward recursions

To derive the forward and backward recursions in this case of partially observable hidden states, we first update the variables of the forward–backward procedure, which incorporates the side information $\mathbf{x}$. The updated forward variable is defined as

$$\overline{\alpha}_t(i)=P(\mathbf{Y}_1^t=\mathbf{y}_1^t,\mathbf{X}_1^t=\mathbf{x}_1^t,Z_t=s_i|\lambda), \tag{4}$$

the probability of observing $(\mathbf{y}_1^t,\mathbf{x}_1^t)$ and being at state $z_t{=}s_i$, given the model λ. Note that z_t is the correct and the underlying hidden state, whereas $\mathbf{x}_1^t$ are the state observations, for which we might have (1) $x_t=s_0$ corresponding to the case that z_t is not actually observed and (2) noisy, if z_t is actually observed. Similarly, the backward variable

$$\overline{\beta}_t(i)=P(\mathbf{Y}_{t+1}^T=\mathbf{y}_{t+1}^T,\mathbf{X}_{t+1}^T=\mathbf{x}_{t+1}^T|Z_t=s_i,\lambda), \tag{5}$$

is the probability of observing $(\mathbf{y}_{t+1}^T,\mathbf{x}_{t+1}^T)$, given the model and the state $z_t{=}s_i$. The updated forward and backward variables are the key variables of incorporating the side information. The following proposition explicitly relates these variables to the side information and provides the corresponding recursions.

Proposition 1. *For the updated forward and backward variables defined in (4) and (5), we have*

$$\overline{\alpha}_t(i)=\nu(x_t,s_i)B_{iy_t}\sum_{j=1}^{N_s}A_{ji}\overline{\alpha}_{t-1}(j),\quad 2\le t\le T,$$

$$\overline{\beta}_t(i)=\sum_{j=1}^{N_s}\nu(x_{t+1},s_j)\overline{\beta}_{t+1}(j)A_{ij}B_{jy_{t+1}},\quad 1\le t\le T-1,$$

where $\nu(x_t,s_i)=1_{\{x_t=s_0\}}(1-\tau)+1_{\{x_t=s_i\}}\tau p+1_{\{x_t\neq s_i\wedge x_t\neq s_0\}}\tau(1-p)/(N_s-1)$, $s_i\neq s_0$, $s_j\neq s_0$.

Proof. Using the marginalization over the random variable Z_{t-1}, we can obtain $\overline{\alpha}_t(i)$ as

$$\overline{\alpha}_t(i)=\sum_{j=1}^{N_s}P(\mathbf{Y}_1^t=\mathbf{y}_1^t,\mathbf{X}_1^t=\mathbf{x}_1^t,Z_t=s_i,Z_{t-1}=s_j|\lambda),$$

which can be expressed, using the product of conditional probabilities, as

$$\overline{\alpha}_t(i)=\sum_{j=1}^{N_s}[P(Y_t=y_t,X_t=x_t,Z_t=s_i|\mathbf{Y}_1^{t-1}=\mathbf{y}_1^{t-1},\mathbf{X}_1^{t-1}$$
$$=\mathbf{x}_1^{t-1},Z_{t-1}=s_j,\lambda)$$
$$P(\mathbf{Y}_1^{t-1}=\mathbf{y}_1^{t-1},\mathbf{X}_1^{t-1}=\mathbf{x}_1^{t-1},Z_{t-1}=s_j|\lambda)].$$

By definition of the updated forward variable, we get

$$\overline{\alpha}_t(i)=\sum_{j=1}^{N_s}P(Y_t=y_t,X_t=x_t,Z_t=s_i|\mathbf{Y}_1^{t-1}=\mathbf{y}_1^{t-1},\mathbf{X}_1^{t-1}$$
$$=\mathbf{x}_1^{t-1},Z_{t-1}=s_j,\lambda)\overline{\alpha}_{t-1}(j),$$

where Markov Property is applied to reach

$$\overline{\alpha}_t(i)=\sum_{j=1}^{N_s}P(Y_t=y_t,X_t=x_t,Z_t=s_i|Z_{t-1}=s_j,\lambda)\overline{\alpha}_{t-1}(j)$$
$$=\sum_{j=1}^{N_s}P(Y_t=y_t,X_t=x_t|Z_t=s_i,Z_{t-1}=s_j,\lambda)P(Z_t$$
$$=s_i|Z_{t-1}=s_j,\lambda)\overline{\alpha}_{t-1}(j)=\sum_{j=1}^{N_s}P(Y_t=y_t,X_t$$
$$=x_t|Z_t=s_i,\lambda)P(Z_t=s_i|Z_{t-1}=s_j,\lambda)\overline{\alpha}_{t-1}(j).$$

Since X_t and Y_t are independent conditioned on (Z_t,λ), we obtain

$$\overline{\alpha}_t(i)=\sum_{j=1}^{N_s}P(Y_t=y_t,X_t=x_t|Z_t=s_i,\lambda)A_{ji}\overline{\alpha}_{t-1}(j)$$
$$=\sum_{j=1}^{N_s}P(X_t=x_t|Z_t=s_i,\lambda)P(Y_t=y_t|Z_t=s_i,\lambda)A_{ji}\overline{\alpha}_{t-1}(j).$$

Then, by definition of the probability of error events in the side information, we get the proposition for the updated forward variable as

$$\overline{\alpha}_t(i)=\nu(x_t,s_i)B_{iy_t}\sum_{j=1}^{N_s}A_{ji}\overline{\alpha}_{t-1}(j),\quad 2\le t\le T.$$

As for the initialization, we set $\overline{\alpha}_1(i)=\nu(x_1,s_i)\pi_iB_{iy_1}$. Similarly, the corresponding recursion for the updated backward variable can be found as

$$\overline{\beta}_t(i)=\sum_{j=1}^{N_s}\nu(x_{t+1},s_j)\overline{\beta}_{t+1}(j)A_{ij}B_{jy_{t+1}},\quad 1\le t\le T-1,$$

for which we have the initialization $\overline{\beta}_T(i)=1$. □

Here, p reflects the confidence that we have on the side information and it is a parameter in our training algorithm. Ideally, when given a set of data, p (named as p_{train} in Section 4) should be set according to the underlying true noise level, $1-p_{\text{true}}$, which is unknown. This brings an immediate tradeoff between setting the confidence too low or too high, when an accurate guess about $1-p_{\text{true}}$ is not present. If we have too high confidence, then our algorithm basically overfits to the noise in the side information, which degrades the state recognition performance as discussed in Section 4. On the other hand, if we have too low confidence, then our algorithm does not fully exploit the side information to its limit. We discuss this later in Section 4, when investigating the robustness of our algorithm to the confidence parameter p (p_{train} in Section 4).

We next present the following proposition that relates $\overline{\epsilon}_t(i,j)$ to the updated forward and backward variables in order to exploit the recursions given in Proposition 1 in the estimation of the HMM parameters in our new framework.

Proposition 2. *With the definitions in (4) and (5), we have*

$$\overline{\epsilon}_t(i,j)=P\big(Z_t=s_i,Z_{t+1}=s_j\big|\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x},\lambda\big)$$

$$= \frac{B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)},$$

where $\nu(x_{t+1},s_j)=1_{\{x_{t+1}=s_0\}}(1-\tau)+1_{\{x_{t+1}=s_j\}}\tau p+1_{\{x_{t+1}\neq s_j \wedge x_{t+1}\neq s_0\}}\tau(1-p)/(N_s-1)$, $s_j \neq s_0$.

Proof. Splitting the observations as $\mathbf{y}=(\mathbf{y}_1^t, y_{t+1}, \mathbf{y}_{t+2}^T)$, and the side information as $\mathbf{x}=(\mathbf{x}_1^t, x_{t+1}, \mathbf{x}_{t+2}^T)$, (3) yields

$$\overline{\epsilon}_t(i,j)=\frac{P(Z_{t+1}=s_j,\mathbf{X}_{t+2}^T=\mathbf{x}_{t+2}^T,\mathbf{Y}_{t+2}^T=\mathbf{y}_{t+2}^T|\lambda)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)} \times P(Z_t=s_i,\mathbf{X}_1^{t+1}=\mathbf{x}_1^{t+1},\mathbf{Y}_1^{t+1}=\mathbf{y}_1^{t+1}|Z_{t+1} = s_j,\mathbf{X}_{t+2}^T=\mathbf{x}_{t+2}^T,\mathbf{Y}_{t+2}^T=\mathbf{y}_{t+2}^T,\lambda).$$

Since $(Z_t,\mathbf{X}_1^{t+1},\mathbf{Y}_1^{t+1})$ is independent with $(\mathbf{X}_{t+2}^T,\mathbf{Y}_{t+2}^T)$ conditioned on (Z_{t+1},λ), we obtain

$$\overline{\epsilon}_t(i,j)=\frac{P(Z_{t+1}=s_j,\mathbf{X}_{t+2}^T=\mathbf{x}_{t+2}^T,\mathbf{Y}_{t+2}^T=\mathbf{y}_{t+2}^T|\lambda)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)} \times P(Z_t=s_i,\mathbf{X}_1^{t+1}=\mathbf{x}_1^{t+1},\mathbf{Y}_1^{t+1}=\mathbf{y}_1^{t+1}|Z_{t+1}=s_j,\lambda),$$

which, re-arranging the conditional probabilities, yields

$$\overline{\epsilon}_t(i,j)=\frac{P(Z_t=s_i,Z_{t+1}=s_j,\mathbf{X}_1^{t+1}=\mathbf{x}_1^{t+1},\mathbf{Y}_1^{t+1}=\mathbf{y}_1^{t+1}|\lambda)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)} \times P\left(\mathbf{X}_{t+2}^T=\mathbf{x}_{t+2}^T,\mathbf{Y}_{t+2}^T=\mathbf{y}_{t+2}^T|Z_{t+1}=s_j,\lambda\right)$$

$$=\frac{P(Z_{t+1}=s_j,X_{t+1}=x_{t+1},Y_{t+1}=y_{t+1}|Z_t=s_i,\mathbf{X}_1^t=\mathbf{x}_1^t,\mathbf{Y}_1^t=\mathbf{y}_1^t,\lambda)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)}$$

$$\times P(Z_t=s_i,\mathbf{X}_1^t=\mathbf{x}_1^t,\mathbf{Y}_1^t=\mathbf{y}_1^t|\lambda)$$
$$\times P(\mathbf{X}_{t+2}^T=\mathbf{x}_{t+2}^T,\mathbf{Y}_{t+2}^T=\mathbf{y}_{t+2}^T|Z_{t+1}=s_j,\lambda).$$

Since $(Z_{t+1},X_{t+1},Y_{t+1})$ and $(\mathbf{X}_1^t,\mathbf{Y}_1^t)$ are independent conditioned on (Z_t,λ), and recognizing the terms $\overline{\alpha}_t(i)$ and $\overline{\beta}_{t+1}(j)$, we obtain

$$\overline{\epsilon}_t(i,j)=\frac{P(Z_{t+1}=s_j,X_{t+1}=x_{t+1},Y_{t+1}=y_{t+1}|Z_t=s_i,\lambda)\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)}$$
$$=\frac{P(X_{t+1}=x_{t+1},Y_{t+1}=y_{t+1}|Z_{t+1}=s_j,Z_t=s_i,\lambda)P(Z_{t+1}=s_j|Z_t=s_i,\lambda)\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)},$$

wherein, Markov Property is used to reach

$$\overline{\epsilon}_t(i,j)=\frac{P(X_{t+1}=x_{t+1},Y_{t+1}=y_{t+1}|Z_{t+1}=s_j,\lambda)P(Z_{t+1}=s_j|Z_t=s_i,\lambda)\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)}.$$

Since X_{t+1} is independent with Y_{t+1} conditioned on (Z_{t+1},λ), we obtain

$$\overline{\epsilon}_t(i,j)=\frac{P(Y_{t+1}=y_{t+1}|Z_{t+1}=s_j,\lambda)P(X_{t+1}=x_{t+1}|Z_{t+1}=s_j,\lambda)P(Z_{t+1}=s_j|Z_t=s_i,\lambda)\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)}.$$

Then, due to the definition of the probability of error event in the side information, we get the proposition as

$$\overline{\epsilon}_t(i,j)=\frac{B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{P(\mathbf{Y}=\mathbf{y},\mathbf{X}=\mathbf{x}|\lambda)}. \quad \square$$

Based on the new set of equations as well as the recursions defined in Proposition 1, we incorporated possibly corrupted side information into the HMM training framework. We finally point out that the forward and backward variables tend to 0 exponentially [15,26] as we are provided longer sequences, i.e., $\overline{\alpha}_T(i)\to 0$, and $\overline{\beta}_1(i)\to 0, \forall i$, as $T\to\infty$. This would create in practice stability issues, i.e., numeric underflow, on any computer if the recursions in Proposition 1 were directly evaluated. Hence, we propose to use the following scaling scheme of [26] in order to avoid such issues. Let us define the normalization factor c_t

$$c_t=\frac{1}{\sum_{i=1}^{N_s}\overline{\alpha}_t(i)} \quad \text{and} \quad \sum_{i=1}^{N_s} c_t\overline{\alpha}_t(i)=1,$$

then we normalize the forward variable $\overline{\alpha}_t(i)$ as $c_t\overline{\alpha}_t(i)$ at each time t after it is calculated with respect to the recursions given in Proposition 1. Similarly, we also normalize the backward variable $\overline{\beta}_t(i)$ with c_t as $c_t\overline{\beta}_t(i)$ at each time t. Then it is easy to see that the re-estimation formulas, i.e., whether they are computed with the normalized or unnormalized forward and backward variables, remain intact. Namely, consider the re-estimation formulas for the state transition probabilities calculated with the normalized forward and backward variables as

$$\hat{A}_{ij}=\frac{\sum_{t=1}^{T-1}\overline{\epsilon}_t(i,j)}{\sum_{t=1}^{T-1}\overline{\gamma}_t(i)} =\frac{\sum_{t=1}^{T-1}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}C_t\overline{\alpha}_t(i)D_{t+1}\overline{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N_s}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}C_t\overline{\alpha}_t(i)D_{t+1}\overline{\beta}_{t+1}(j)},$$

where $C_t=\prod_{i=1}^t c_i$ and $D_{t+1}=\prod_{i=t+1}^T c_i$. Hence, noting that $C_tD_{t+1}=\prod_{i=1}^T c_i, \forall t$,

$$\hat{A}_{ij}=\frac{\sum_{t=1}^{T-1}\overline{\epsilon}_t(i,j)}{\sum_{t=1}^{T-1}\overline{\gamma}_t(i)}$$
$$=\frac{\prod_{i=1}^T c_i\sum_{t=1}^{T-1}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{\prod_{i=1}^T c_i\sum_{t=1}^{T-1}\sum_{j=1}^{N_s}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}$$
$$=\frac{\sum_{t=1}^{T-1}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N_s}B_{jy_{t+1}}\nu(x_{t+1},s_j)A_{ij}\overline{\alpha}_t(i)\overline{\beta}_{t+1}(j)}.$$

Similarly, the estimates for the conditional observation probabilities $\hat{B}_{ij}$ as well as the initial state probabilities $\hat{\pi}_i$ also remain exact with this normalization scheme. Therefore, this normalization provides the numerical stabilization of the proposed estimation method. In the next section, we provide examples that demonstrate the performance of the new set of training updates under different scenarios.

4. Simulations

In this section, we demonstrate the performance of our method through simulations using data generated with the following HMM parameters:

$$N_s = 3, \quad N_v = 3, \quad \pi = [0.3 \;\; 0.3 \;\; 0.4], \quad \text{and}$$

$$A = \begin{bmatrix} 0.8 & 0.19 & 0.01 \\ 0.01 & 0.8 & 0.19 \\ 0.19 & 0.01 & 0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0.1 & 0.6 \end{bmatrix}.$$

For these simulations, we generate a test sequence of length 500 and a training sequence of length 250 along with the side information of a relatively high noise level, $1-p_{\text{true}} = 0.4$, and a relatively low noise level, $1-p_{\text{true}} = 0.2$, with τ ranging from 0 to 0.6. We emphasize that the exact noise level may not be known by the algorithm. Hence, we provide p_{train} to the algorithm which may not be equal to the p_{true}. Here, the parameter p_{train} reflects the confidence (equivalently the expected noise level) that we have on the side information. Since this confidence on the side information might not be accurate, i.e., p_{train} does not necessarily match with p_{true}, for analyzing the sensitivity of our method to the confidence parameter, we train our algorithm with different choices for p_{train}: (1) we set confidence that is in the proximity of p_{true} ($p_{\text{train}} \sim p_{\text{true}}$), i.e., $p_{\text{train}} \in \{0.55, 0.6, 0.65\}$, if $p_{\text{true}} = 0.6$ and $p_{\text{train}} \in \{0.75, 0.8, 0.85\}$, if $p_{\text{true}} = 0.8$, (2) we set too high confidence on the side information ($p_{\text{train}} \gg p_{\text{true}}$), i.e., $p_{\text{train}} = 1$, when $p_{\text{true}} \in \{0.6, 0.8\}$ and, (3) we set too low confidence ($p_{\text{train}} \ll p_{\text{true}}$), i.e., $p_{\text{train}} = 0.5$, when $p_{\text{true}} = 1$. Using the training sequence, we first estimate the unknown model parameters, A_{ij}, B_{ij}, and π_{ij}. Then, on the test sequence, the hidden state sequence is estimated by the Viterbi algorithm [27,28] using the estimated model parameters. We apply our method on 500 different pairs of test and training sequences and we present the resulting average state recognition error rates for all the cases aforementioned. In order to show the efficacy of incorporating the side information by our method, we compare the state recognition error rates of our algorithm with (1) Baseline Performance, the state recognition error rate if the model parameters are estimated by the ordinary HMM parameter estimation. This is the performance, which is readily achievable with no side information. (2) The Oracle, the state recognition error rate if the true model parameters are directly used in the state estimation on the test sequence. This is the performance limit if the HMM training algorithm is run on infinite amount of training data, which is only asymptotically achievable. (3) Limit of Algorithm, the state recognition error rate if the side information is completely accurate and the algorithm is trained with complete confidence on the side information, i.e., $p_{\text{true}} = 1$, $p_{\text{train}} = 1$. Finally, this is the performance limit that our algorithm can gain at most by exploiting the side information. Here, we name the difference between the Baseline Performance and the Oracle as the "achievable margin" since no algorithm can obtain state recognition improvements more than the achievable margin, provided that, as in this work, first the model parameters are estimated and then used in the Viterbi algorithm for state recognition.

Our simulations show that the performance of our method, provided that $p_{\text{train}} \sim p_{\text{true}}$, improves with the amount of side information that is indicated by τ. In particular, when we have accurate access to the hidden states, i.e., $p_{\text{true}} = p_{\text{train}} = 1$, the state recognition rate in the test sequence, labeled as Limit of Algorithm in Fig. 2, consistently approaches to the Oracle as τ increases and reaches ~90% gain (the performance improvement over the baseline corresponds to ~90% of the achievable margin) with 30% additional information on states, i.e., $\tau = 0.3$, as shown in Fig. 2. This proves the efficacy of our method with incorporating the side information. On the other hand, in the case of noisy access to the hidden states such that 20% of the state observations are mislabeled, i.e., $p_{\text{true}} = 0.8$, our method (when $p_{\text{train}} \sim p_{\text{true}}$) is able to provide substantial gain, 70%, at $\tau = 0.3$. In this case, as τ increases, the recognition approaches to Limit of Algorithm showing that our algorithm optimally incorporates the side information under noise asymptotically. Even if the noise level is further increased up to a level as high as 40% mislabeling, we still obtain a gain that consistently increases with τ, when $p_{\text{train}} \sim p_{\text{true}}$. Thus, our method is robust to noise. Nevertheless, the algorithm must not rely on the side information with too high confidence. Specifically, when we have the confidence $p_{\text{train}} = 1$ in case of high noise level,

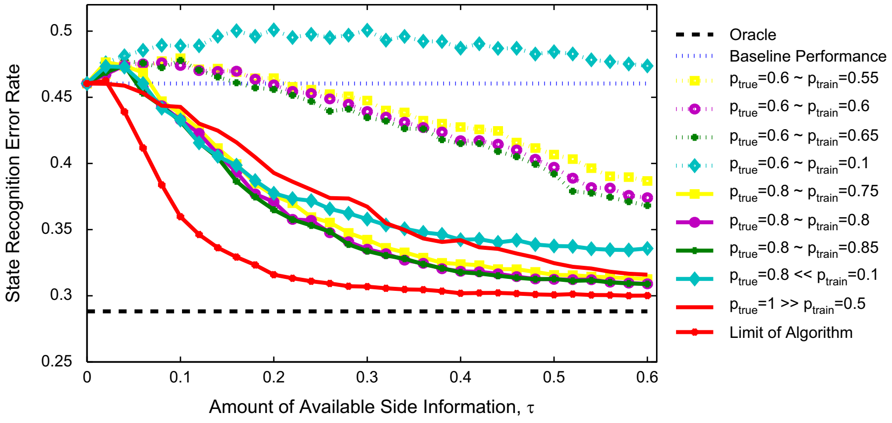

Fig. 2. Simulation results for different scenarios. Our algorithm is trained with $p_{\text{train}} \in \{0.55, 0.60, 0.65\}$ when $p_{\text{true}} = 0.60$ and $p_{\text{train}} \in \{0.75, 0.80, 0.85\}$ when $p_{\text{true}} = 0.80$. The State Recognition Error Rates are estimated by the Viterbi algorithm. Performance of our algorithm is compared against three performance limits: (1) Baseline Performance, error rate by ordinary HMM using no side Information, (2) Oracle, error rate if the true model parameters are used in state recognition, and (3) Limit of Algorithm, $p_{\text{train}} = p_{\text{true}} = 1$. See the text for details.

i.e., $p_{\text{true}} = 0.6$, we do not obtain any improvement compared to the baseline. On the contrary, the algorithm does not fully exploit the side information to its limit, if the confidence is too low. For instance, in case of $p_{\text{train}} = 0.5$ and $p_{\text{true}} = 1$, the rate of performance improvement with τ is significantly slower than that of Limit of Algorithm, i.e., $p_{\text{true}} = 1, p_{\text{train}} = 1$. According to our simulations, setting the confidence in the proximity of the true noise level is sufficient to obtain the maximum gain, i.e., our algorithm does not require an exact match between p_{true} and p_{train}. This demonstrates that our algorithm is also robust to the mismatches in the confidence parameter.

5. Conclusion

In this paper, we introduced a new parameter estimation algorithm for HMM, when we have partial and noisy access to the hidden state sequence as side information. This side information can be seen as partial labeling, "possibly wrong", of the hidden states. In this work, we model mislabeling and partial labeling of the hidden states jointly in one complete framework. This framework naturally recovers the unsupervised HMM training if the partial access to the hidden states is turned off. In our simulations, we observed that, using this side information, we considerably improved the state recognition performance, up to 70%, with respect to the "achievable margin". Moreover, our method is shown to be robust to the training conditions. Finally, since this framework includes possible mislabeling events, our algorithm models realistic training conditions more accurately than the ordinary HMM training. Hence, we expect the same performance improvement in other examples.

References

[1] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 257–286.

[2] K.Y. Lee, J. Lee, A study on IMM with NPHMM and an application to speech enhancement, Signal Processing 84 (2004) 1701–1707.

[3] K.Y. Lee, J. Rheem, Smooth approach using forward–backward Kalman filter with Markov switching parameters for speech enhancement, Signal Processing 80 (2000) 2579–2588.

[4] D.X. Sun, L. Deng, C.F.J. Wu, State-dependent time warping in the trended hidden Markov model, Signal Processing 39 (1994) 263–275.

[5] M.D. Moore, M.I. Savic, Speech reconstruction using a generalized HSMM (GHSMM), Digital Signal Processing 14 (2004) 37–53.

[6] D. Cutting, J. Kuipec, J. Pedersen, P. Sibun, A practical part-of-speech tagger, in: Proceedings of the Third Conference on Applied Natural Language Processing, 1992, pp. 133–140.

[7] P.C. Woodland, D. Povey, Large scale discriminative training of hidden Markov models for speech recognition, Computer Speech and Language 16 (2002) 25–47.

[8] S. Kozat, K. Visweswariah, R. Gopinath, Efficient, low latency adaptation for speech recognition, in: IEEE International Conference on Acoustic Speech and Signal Processing, 2007, pp. 777–780.

[9] E. Birney, Hidden Markov models in biological sequence analysis, IBM Journal of Research and Development 45 (2001) 449.

[10] V. Fonzo, F. Aluffi-Pentini, V. Parisi, Hidden Markov models in bioinformatics, Current Bioinformatics (2007) 49–61.

[11] L. Bordes, P. Vandekerkhove, Statistical inference for partially hidden Markov models, Communications in Statistics 34 (2005) 1081–1104.

[12] B. Merialdo, Tagging english text with a probabilistic model, Computational Linguistics 20 (1994) 155–171.

[13] D. Elworthy, Does Baum–Welch re-estimation help taggers?, in: Proceedings of the Fourth Conference on Applied Natural Language Processing, ANLC '94, 1994, pp. 53–58.

[14] K. Seymore, A. Mccallum, R. Rosenfeld, Learning hidden Markov model structure for information extraction, in: AAAI 99 Workshop on Machine Learning for Information Extraction, 1999, pp. 37–42.

[15] Y. Ephraim, N. Merhav, Hidden Markov processes, IEEE Transactions on Information Theory 48 (2002) 1518–1569.

[16] K. Thulasiraman, M.N.S. Swamy, Graphs: Theory and Algorithms, John Wiley and Son, 1992.

[17] E. Baccarelli, R. Cusani, Recursive Kalman-type optimal estimation and detection of hidden Markov chains, Signal Processing 51 (1) (1996) 55–64.

[18] T. Scheffer, S. Wrobel, Active learning of partially hidden Markov models, in: Proceedings of ECML/PKDD Workshop on Instance Selection, 2001.

[19] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley Series in Probability and Statistics, Wiley-Interscience, 2008.

[20] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society 39 (1977) 1–38.

[21] L.E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, Annals of Mathematical Statistics 41 (1970) 164–171.

[22] S. Forchhammer, J. Rissanen, Partially hidden Markov models, IEEE Transactions on Information Theory 42 (1996) 1253–1256.

[23] O. Chapelle, B. Schlkopf, A. Zien, Semi-Supervised Learning, Adaptive Computation and Machine Learning, MIT Press, 2006.

[24] L.E. Baum, J.A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bulletin of the American Mathematical Society 73 (1967) 360–363.

[25] L.E. Baum, G.R. Sell, Growth functions for transformations on manifolds, Pacific Journal of Mathematics 27 (1968) 211–227.

[26] S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, Bell System Technical Journal 62 (4) (1983).

[27] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory 13 (1967) 260–269.

[28] G.D. Forney, The Viterbi algorithm, Proceedings of the IEEE 61 (1973) 268–278.