Identification of Hidden Markov Models Using Spectral Learning with Likelihood Maximization

Robert Mattila, Cristian R. Rojas, Vikram Krishnamurthy and Bo Wahlberg

Abstract— **In this paper, we consider identifying a *hidden Markov model* (HMM) with the purpose of providing estimates of joint and conditional (posterior) probabilities over observation sequences. The standard procedure, i.e., the Baum-Welch/expectation-maximization algorithm, has recently been challenged by methods of moments. Such methods employ low-order moments to provide parameter estimates and come with several benefits, e.g., consistency and low computational cost. This paper focuses on a particular method that identifies an *observable representation* of an HMM. We aim to reduce the gap in statistical efficiency that results from restricting to only low-order moments in the training data. In particular, we propose improving the initial estimates by approximately maximizing the associated likelihood function as a second step in the estimation procedure. The maximization is performed by employing a second order optimization procedure. We demonstrate an improved statistical performance using the proposed algorithm in numerical simulations.**

I. Introduction

Numerous applications in signal processing and control rely on the *hidden Markov model* (HMM): computational biology and genomic sequence analysis [1], [2], automatic speech recognition [3], social network analysis [4], financial modeling [5], etc. At its core, the HMM is a stochastic model where discrete observations are made independently conditioned on a discrete latent state that evolves over time according to Markovian dynamics. For the purpose of control and filtering, the model is usually built from data, i.e., identified, since first principles modelling can be prohibitively laborious.

The most widely used identification procedures for HMMs employ *maximum likelihood* (ML) techniques. They are commonly based on iterative hill-climbing algorithms such as the *expectation-maximization* (EM, [6]) algorithm, or Newton based methods. However, due to the non-convexity of the likelihood function and the local-search nature of these algorithms, they are prone to converge to local minima.

Recently, several method of moments estimators (see, e.g., [7] for an introduction in the general setting) have been proposed for HMMs [8]–[15]. These estimators first compute empirical estimates of low-order moments given an HMM observation sequence. They then use (inverted) relations between these moments and the system parameters to provide parameter estimates. Benefits of these methods over local search procedures (e.g., EM) include lower computational cost, and that they sometimes are amenable to statistical guarantees in terms of consistency and finite-sample bounds.

However, method of moments algorithms that consider only lower order moments suffer from a loss of statistical efficiency. Essentially, only short substrings of the full data sequence are considered in the estimation procedure, and hence, a lot of the information available in correlations is lost.

In this paper, we focus on a popular method proposed in [8]: the *spectral learning* (SL) algorithm. SL identifies an *observation operator representation* (OOR, [16]) of the HMM, which allows computation of filtering and prediction quantities related to observations. SL is computationally very efficient – only one pass over the data is required to obtain moment estimates, and the ensuing moment-matching is solved using simple linear algebra routines. The main aim of this paper is to reduce the aforementioned gap in statistical efficiency for SL. We propose a method of improving the estimated model by considering the statistics of the full data as a second step – allowing for extraction of more information.

In particular, we use the initial OOR resulting from SL to approximately compute the likelihood with respect to the observed data. This gives us a proxy to the likelihood function. Similar to the *sensitivity equations* of the HMM filter, that can be used to compute the gradient and Hessian of the likelihood in the standard parametrization (see, e.g., [4], [17]), we obtain the gradient and Hessian of the approximate likelihood. These allow us to employ a second order optimization procedure, inspired by the *indirect prediction-error method* [18], used in linear system identification, to improve the estimates.

In summary, the main contributions of this paper are:

- an identification method of incorporating the statistics of the full observation sequence as a second step in SL, hence exploiting more of the information available in the data and increasing the statistical performance;
- a demonstration of the performance of the proposed method in numerical simulations.

The outline of the rest of the paper is as follows. We first provide a brief overview of related work in Section I-A. Section II then details the necessary notation, the background related to HMMs and the OOR formulation, as well as the problem setup. The proposed method is given in Section III. The following section, Section IV, discusses limitations and practical considerations of the algorithm. Section V presents

This work was partially supported by the Swedish Research Council under contract 2016-06079 and the Linnaeus Center ACCESS at KTH. Robert Mattila, Cristian R. Rojas and Bo Wahlberg are with the Department of Automatic Control and ACCESS, School of Electrical Engineering, KTH Royal Institute of Technology. Stockholm, Sweden. (e-mails: rmattila,crro,bo@kth.se). Vikram Krishnamurthy is with the Department of Electrical and Computer Engineering, Cornell University. Cornell Tech, NY, USA. (e-mail: vikramk@cornell.edu).

several numerical evaluations of the proposed procedure. The paper is concluded with a brief discussion in Section VI.

A. Related work

HMM identification is by now a classical field with a vast literature. ML estimation using the now-standard Baum-Welch algorithm (EM applied to HMMs) is presented and discussed in, e.g., [3], [4] and [17]. Several alternative methods have been proposed due to its drawbacks related to only local convergence and high computational cost. Methods inspired by ideas from subspace identification for linear systems (see, e.g., [19]) include [20]–[25].

Related to these, an interest for utilizing method of moments has in the past few years spread in the machine learning community. Such methods (of moments) that recover the standard parameters (i.e., the transition and observation matrices) of an HMM include [8, Appendix A], which has been further generalized in [9] and [10]; and [11]–[13].

For certain tasks, learning a full representation of an HMM is superfluous. Methods of moments have therefore also recently been formulated to learn observable representations (e.g., [16], [26]) of HMMs, which avoid estimating the transition and observation matrices explicitly – see e.g., [8], [14], [15]. In terms of increasing the statistical efficiency of such methods, [27] and [28] consider initializing EM in the resulting estimates. The authors of [29] reformulate SL [8] in a *generalized method of moments* (GMM) framework that allows for iterative reweighing of the estimated moments. The distinguishing feature of our work is that we propose a second order scheme.

II. Preliminaries and Problem Formulation

Element i of a vector is denoted $[\cdot]_i$, and the element at row i and column j of a matrix is denoted $[\cdot]_{ij}$. Vectors are column vectors, unless transposed. We let $\mathbb{1}$ denote the vector of ones and e_i the ith Euclidean standard-basis vector. The operation $\operatorname{diag}(\cdot)$ returns a matrix with the vector $\cdot$ on the diagonal. For brevity, we sometimes shorten a sequence $y_1, \dots, y_k$ to $y_{1:k}$. The Moore-Penrose matrix pseudo-inverse is denoted †.

A. Hidden Markov models

We consider finite-state finite-observation alphabet *hidden Markov models* (HMMs): $\{\mathcal{X}, \mathcal{Y}, P, B, \pi_0\}$, where $\mathcal{X}$ is a discrete state-space of X states; $\mathcal{Y}$ is a discrete observational alphabet of size Y; π_0 an initial distribution over $\mathcal{X}$; and, $P \in \mathbb{R}^{X\times X}$ and $B \in \mathbb{R}^{X\times Y}$ are row-stochastic transition and observation matrices defined, respectively, as

$$[P]_{ij} = \Pr[x_{k+1} = j | x_k = i], \tag{1}$$

and

$$[B]_{ij} = \Pr[y_k = j | x_k = i]. \tag{2}$$

Assuming the transition matrix P to be ergodic (i.e., aperiodic and irreducible), denote by π_∞ its stationary distribution.

B. Problem formulation

The problem we consider is motivated by the common situation where predicting the future value of a (non-deterministic) quantity is of interest. Formally, we want to solve:

Problem 1. *Consider a given sequence of N observations sampled from an HMM with unknown transition and observation matrices. For an arbitrary observation y_{k+1} and an arbitrary sequence of observations $y_{1:k}$, provide a method to estimate the joint and conditional probabilities of these sequences in the unknown HMM, i.e., estimate $\Pr[y_{1:k}]$ and $\Pr[y_{k+1}|y_{1:k}]$.*

The standard approach, i.e., EM, solves the problem in two steps. The EM algorithm alternates between i) estimating the corresponding non-observed latent states, and ii) fitting the HMM parameters P and B to the estimated state sequence and the observed output sequence. An estimated model is available once a convergence criterion is fulfilled. The (estimated) joint probability can then be computed directly. For the posteriors, an HMM filter can be employed to obtain belief states over the latent variables. These can then be propagated to obtain the sought posteriors. The next subsection outlines how the problem is solved using SL.

C. Observable operator representation (OOR) of an HMM

In comparison to EM, SL avoids identifying the transition and observation matrices separately when solving Problem 1. Instead, it identifies a parametrization of the HMM (roughly, in terms of products between P and B) which can be related to only observable quantities – removing the need to estimate the latent state sequence. In particular, define the first, second and third order moments of the HMM as,

$$[M_1^k]_i = \Pr[y_k = i], \tag{3a}$$

$$[M_{2,1}^k]_{ij} = \Pr[y_{k+1} = i, y_k = j], \tag{3b}$$

$$[M_{3,y,1}^k]_{ij} = \Pr[y_{k+2} = i, y_{k+1} = y, y_k = j], \tag{3c}$$

respectively. In the limit, as $k \to \infty$, or in the case that $\pi_0 = \pi_\infty$, we obtain the stationary moments (that can be directly estimated from observed data):

$$M_1 \overset{\text{def.}}{=} M_1^\infty,\ M_{2,1} \overset{\text{def.}}{=} M_{2,1}^\infty,\ M_{3,y,1} \overset{\text{def.}}{=} M_{3,y,1}^\infty. \tag{4}$$

It is shown in [8] that these quantities are sufficient to estimate $\Pr[y_{k+1}|y_{1:k}]$ and $\Pr[y_{1:k}]$ consistently under fairly general assumptions on the HMM: P and B are full rank, and $\pi_\infty > 0$. In particular, it can be done as follows. First, it is required to find a matrix $U \in \mathbb{R}^{Y\times X}$ such that the product BU is invertible.[1] The following *observation operator representation* (OOR) of the HMM is then defined (recall that † denotes the Moore-Penrose pseudo-inverse) as

$$b_1 = U^T M_1, \tag{5a}$$

$$b_\infty = (M_{2,1}^T U)^\dagger M_1, \tag{5b}$$

$$\mathcal{B}_y = U^T M_{3,y,1} (U^T M_{2,1})^\dagger. \tag{5c}$$

[1] This is usually done by taking U to be the left singular vectors of the thin SVD of $M_{2,1}$ – see [8, Lemma 2].

Note that b_1 and b_∞ are vectors, and $\{\mathcal{B}_y\}_{y=1}^{Y}$ is a set of Y matrices. In terms of this parametrization, the joint probability of a sequence can be computed (see [8, Lemma 3]) as

$$\Pr[y_1, \ldots, y_k] = b_\infty^T \mathcal{B}_{y_k} \ldots \mathcal{B}_{y_1} b_1. \tag{6}$$

Conditional probabilities, i.e., output posteriors, can be obtained (see [8, Lemma 4]) by introducing an "internal state"

$$b_{k+1} = \frac{\mathcal{B}_{y_k} b_k}{b_\infty^T \mathcal{B}_{y_k} b_k}, \tag{7}$$

and then computing

$$\Pr[y_k | y_1, \ldots, y_{k-1}] = b_\infty^T \mathcal{B}_{y_k} b_k. \tag{8}$$

The SL procedure consists of using empirical estimates of the moments (4) to generate an estimate of the OOR (5a)-(5c). These estimates are then used in (6), (7) and (8) to, in turn, compute estimates $\widehat{\Pr}[y_{1:k}]$ and $\widehat{\Pr}[y_k|y_{1:k-1}]$.[2] The usual assumption, to facilitate finite-sample analysis, is that triplets of observations are sampled independently to form moment estimates. However, in practice, the full observation sequence is employed to estimate the moments:

$$[\hat{M}_{3,y,1}]_{ij} = \frac{1}{N-2} \sum_{k=1}^{N-2} \mathrm{I}\{y_{k+2} = i, y_{k+1} = y, y_k = j\}, \tag{9}$$

and similarly for $\hat{M}_{2,1}$ and $\hat{M}_1$. The main idea of this paper is to exploit this by extracting more of the information available in the observed data $y_{1:k}$ than just that of third order correlations employed in the estimator (9).

III. Spectral Learning with Likelihood Maximization

In this section, we outline the proposed method that improves upon the solution (5a)-(5c) and (6)-(9) provided to Problem 1 by SL.

A. Maximum likelihood estimation

In ML estimation, the unknown quantities θ parametrizing the distribution of the data are chosen as to maximize the likelihood of the observed data $y_{1:N}$:

$$\hat{\theta}_{\text{ML}} = \arg\max_{\theta \in \Theta} \Pr[y_1, \ldots, y_N; \theta] \overset{\text{def.}}{=} \arg\max_{\theta \in \Theta} \mathcal{L}_N(\theta), \tag{10}$$

where Θ is the feasible parameter set. Note that this naturally allows the statistics of the full data sequence to be exploited. In contrast, the empirical estimator (9) of the moments $M_{3,y,1}$ utilizes the full data sequence, but effectively, only substrings of length three. Moreover, the ML estimate $\hat{\theta}_{\text{ML}}$ has many attractive statistical properties. We propose combining SL with likelihood maximization in order to obtain the advantages of both approaches.

As mentioned in the previous section, it is shown in [8] that the joint probability of a sequence of observations can be computed using an OOR as

$$\Pr[y_1, \ldots, y_k] = b_\infty^T \mathcal{B}_{y_k} \ldots \mathcal{B}_{y_1} b_1. \tag{11}$$

If the sequence $y_{1:k}$ is taken to be the observed data then the likelihood is computed. Below, we will discuss how a hill-climbing optimization algorithm can then be used to maximize this expression with respect to the parameters

$$\theta \overset{\text{def.}}{=} \{b_1, b_\infty, \{\mathcal{B}_y\}_{y=1}^{Y}\}, \tag{12}$$

subject to suitable constraints – effectively solving problem (10) locally.

However, calculating the likelihood of the observed data using relation (11) would quickly result in a numerical underflow. As is customary, we therefore chose to work with the log-likelihood $l_N(\theta) \overset{\text{def.}}{=} \ln \mathcal{L}_N(\theta)$ instead. In general, we have that

$$l_N(\theta) = \sum_{k=1}^{N} \ln \Pr[y_k | y_{1:k-1}], \tag{13}$$

by repeated conditional factorization. Each term in (13) can be expressed using (8), so that

$$l_N(\theta) = \sum_{k=1}^{N} \ln b_\infty^T \mathcal{B}_{y_k} b_k, \tag{14}$$

where the b_k:s are computed recursively using (7).

In summary, we can calculate the log-likelihood $l_N(\theta)$ using the recursive procedure:

$$\begin{aligned} l_{k+1}(\theta) &= l_k(\theta) + \ln b_\infty^T \mathcal{B}_{y_k} b_k, \\ b_{k+1} &= \frac{\mathcal{B}_{y_k} b_k}{b_\infty^T \mathcal{B}_{y_k} b_k}, \end{aligned} \tag{15}$$

for $k = 1, \ldots, N-1$, with $l_1(\theta) = \ln b_\infty^T \mathcal{B}_{y_1} b_1$.

B. Likelihood improving step

We will now describe how an initial set of parameters θ_{init} can be improved by means of ML estimation. Assume for now – we provide details in Section IV-B – that the gradient and Hessian, $g \overset{\text{def.}}{=} \nabla_\theta l_N(\theta_{\text{init}})$ and $H \overset{\text{def.}}{=} \nabla_\theta^2 l_N(\theta_{\text{init}})$, respectively, of the log-likelihood (14) can be computed. We first make a second order approximation of the likelihood function around θ_{init},

$$\begin{aligned} l_N(\theta) \approx\ & l_N(\theta_{\text{init}}) \\ & + g^T(\theta - \theta_{\text{init}}) + \frac{1}{2}(\theta - \theta_{\text{init}})^T H (\theta - \theta_{\text{init}}). \end{aligned} \tag{16}$$

This approximation is used sequentially as a surrogate for $l_N(\theta)$ in the Newton-Raphson method of optimization. Finding a stationary point, i.e., maximizing if H is negative definite, yields a standard Newton-Raphson update (see, e.g., [30]) as

$$\theta_{\text{NR}} = \theta_{\text{init}} - [H]^{-1} g. \tag{17}$$

However, this relies on the assumption that θ is unconstrained, which is not the case in our setting.

Inspired by the methodology used in the *indirect prediction-error method* [18], we propose that the parameters are updated as follows. Again, the second-order approximation is used as a surrogate for $l_N(\theta)$, but the constraints $\theta \in \Theta$ are imposed to retain the feasibility of the parameters,

$$\max_{\theta} \quad g^T(\theta - \theta_{\text{init}}) + \frac{1}{2}(\theta - \theta_{\text{init}})^T H (\theta - \theta_{\text{init}})$$

[2] A normalization factor is needed in (8) when computing with the estimated OOR. However, it is argued (in [8]) that this factor is always close to one.

$$\text{s.t.} \quad \theta \in \Theta, \tag{18}$$

which reduces to the standard Newton-Raphson update (17) for unconstrained θ. (The constant term has been dropped since it does not influence the optimization problem.)

A challenging complication is that in terms of θ, as defined in (12), the set Θ is difficult to specify. For example, to guarantee that the new iterate yields valid quantities with respect to relation (6), we have to enforce constraints of the type

$$\Pr[y_{1:k}] = b_\infty^T \mathcal{B}_{y_k} \ldots \mathcal{B}_{y_1} b_1 \geq 0, \tag{19}$$

for all horizon lengths k and sequences $y_{1:k}$, as well as

$$\sum_{y_{1:k}} \Pr[y_{1:k}] = \sum_{y_{1:k}} b_\infty^T \mathcal{B}_{y_k} \ldots \mathcal{B}_{y_1} b_1 = 1, \tag{20}$$

for all horizon lengths k. This infinite number of constraints is non-trivial to enforce in the optimization update (18) – in fact, it is a well-known, and difficult, problem to ensure that empirical OORs fulfill such constraints, see, e.g., [26], [31].

C. Re-parametrized optimization

To circumvent these difficulties, we choose as decision variables in the optimization problem (18) the moments of the HMM instead. Let the new parameter vector be $\mu = \{M_{3,y,1}\}_{y=1}^{Y}$ (since the lower order moments can be calculated by marginalization), and note that $\theta = \theta(\mu)$ since the OOR θ is obtained using the mappings (5a)-(5c) of the moments μ.

In this parametrization, some of the necessary constraints are easy to formulate. First of all, we require the elements to fulfill probability-type constraints: non-negativity,

$$[M_{3,y,1}]_{ij} \geq 0, \forall y \in \mathcal{Y}, \forall i,j \in \mathcal{Y}, \tag{21}$$

and sum-to-one,

$$\sum_{y=1}^{Y} \sum_{i,j=1}^{Y} [M_{3,y,1}]_{ij} = 1. \tag{22}$$

Secondly, since the moments are stationary, we have (see, e.g., [32]) that

$$\sum_{z=1}^{Y} [M_{3,i,1}]_{zj} = \sum_{z=1}^{Y} [M_{3,j,1}]_{iz}, \tag{23}$$

for all $i, j \in \mathcal{Y}$. This means that if we marginalize out either y_k or y_{k+2} we get the same distribution over the remaining two indices.

Denote the set defined by these constraints as $\mathcal{M}$. Then the parameter update (18) can be relaxed into the two steps: i) solve

$$\begin{aligned} \max_{\mu} \quad & \bar{g}^T(\mu - \mu_{\text{init}}) + \frac{1}{2}(\mu - \mu_{\text{init}})^T \bar{H} (\mu - \mu_{\text{init}}) \\ \text{s.t.} \quad & \mu \in \mathcal{M}, \end{aligned} \tag{24}$$

where $\bar{g} \overset{\text{def.}}{=} \nabla_\mu l_N(\theta(\mu_{\text{init}}))$ and $\bar{H} \overset{\text{def.}}{=} \nabla_\mu^2 l_N(\theta(\mu_{\text{init}}))$, and ii) map the updated moments μ into an OOR using (5a)-(5c).

This is a relaxation because $\mu \in \mathcal{M}$ does not imply that $\theta(\mu) \in \Theta$ (e.g., the OOR $\theta(\mu)$ might yield negative probabilities when used in relation (6)). However, compared to the conditions fulfilled by the initial empirical OOR (resulting from mapping the estimator (9) using (5a)-(5c)), it is not a relaxation. In fact, these constraints are stricter since the empirical moment-estimates are not guaranteed to be stationary (and hence, to lie in $\mathcal{M}$) for finite data. That the initial empirical OOR does necessarily lie in Θ will be discussed below, in Section IV-C.

IV. Practical Considerations and Limitations

In this section, we discuss some practical considerations and limitations of the method outlined in the previous section.

A. Convexity

Solving a general *quadratic program* (QP) is known to be a difficult problem. If the second-order term is indefinite – in fact, a single eigenvalue is sufficient – the problem is NP-hard [33]. On the other hand, convex QPs can be solved efficiently. The constraints in problem (24) are clearly convex. With regards to the cost function, it is not necessary that the quadratic term $\bar{H}$ is completely semidefinite – only that its restriction to the feasible set is.

It is possible to make this explicit (see, e.g., [30]). Write the equality constraints (22) and (23) as linear equations $A\mu = b$, where one row of A and one row of b are simply ones, corresponding to (22), and the other rows have alternating positive and negative ones and zeros, corresponding to (23). Then re-parametrize the problem in terms of the null-base N of the coefficient matrix A as $\mu - \mu_{\text{init}} = N\alpha$, where α are new decision variables.

Problem (24) can then be written, with this explicit restriction to the feasible set, as

$$\begin{aligned} \max_{\alpha} \quad & \bar{g}^T N\alpha + \frac{1}{2}\alpha^T N^T \bar{H} N \alpha \\ \text{s.t.} \quad & N\alpha \geq -\mu_{\text{init}}, \end{aligned} \tag{25}$$

where the inequality is interpreted elementwise. The moments are updated by taking $\mu = \mu_{\text{init}} + N\alpha$, and the corresponding OOR $\theta(\mu)$ is obtained by computing the mappings (5a)-(5c). With enough data, the initial empirical estimate is expected to be in a neighbourhood of the maximum of the likelihood, so that $N^T \bar{H} N$ is negative definite and convex optimization methods can be employed to solve the problem efficiently.

B. Computing the gradient and Hessian

Problem (24) requires that the gradient and Hessian of the log-likelihood (14) can be computed. We provide two methods of obtaining these quantities.

1) Direct computation: It is possible to derive recursive equations in a similar fashion as to how recursive equations for the gradient and Hessian of the likelihood for an HMM parametrized in the standard manner (i.e., using P and B) are derived. This is, roughly, done by differentiating the HMM filter, yielding the *sensitivity equations* – see, e.g., [4] or [17].

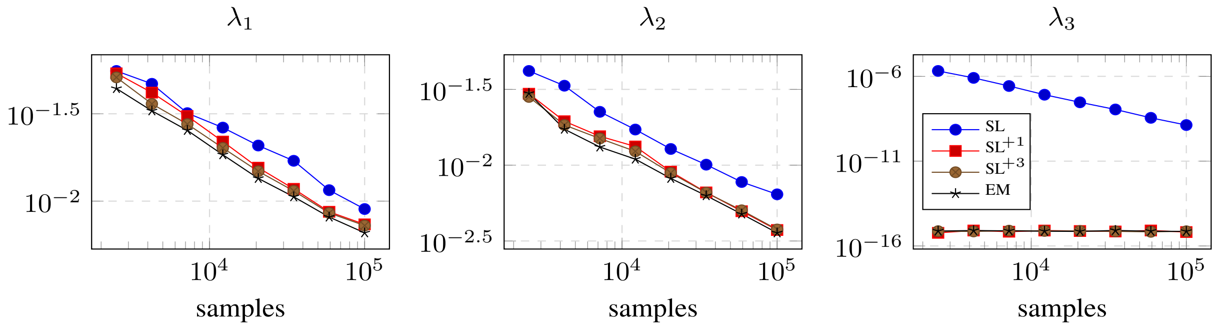

Fig. 1: RMSE of the three eigenvalues of $\hat{\mathcal{B}}$ compared to those of the true transition matrix P. $\hat{\mathcal{B}}$ was computed using spectral learning (SL); one (SL^{+1}) and three (SL^{+3}) subsequent iterations of the proposed method. Also shown are the errors using EM started in the true parameters (EM).

In our setting, one would first differentiate the recursion (15) with respect to θ and then use the chain rule to obtain the derivatives with respect to μ. However, these expressions quickly become tedious to implement.

2) Automatic differentiation (AD): An alternative to the explicit calculation is AD – see, e.g., [34] for a complete treatment. Both the accuracy and the computational complexity of AD are attractive. AD can compute derivatives of arbitrary order that are accurate to working precision. Moreover, the computational complexity of evaluating the gradient of a function is proportional to the complexity of evaluating the function itself. Evaluating the recursion (15) requires $\mathcal{O}(NX^2)$ operations, and hence, so does acquiring the gradient. This is comparable to one iteration of EM. AD was used in the simulations in Section V.

C. Approximate likelihood

In practice, empirical estimates are used of the HMM moments and, in turn, the OOR parametrization θ. This results in a few subtleties. The likelihood calculation (14) builds on the relation (6), which is valid when θ corresponds to an HMM. However, with empirical estimates, we have that

$$\begin{aligned}\hat{b}_\infty^T \hat{\mathcal{B}}_{y_N} \dots \hat{\mathcal{B}}_{y_1} \hat{b}_1 &= \widehat{\Pr}[y_1, \dots, y_N] \\ &\approx \Pr[y_1, \dots, y_N] \\ &= b_\infty^T \mathcal{B}_{y_N} \dots \mathcal{B}_{y_1} b_1. \qquad (26)\end{aligned}$$

This means that the recursive procedure (15) does not necessarily calculate a log-likelihood. In particular, $\widehat{\Pr}[y_{1:N}]$ is not even guaranteed to be non-negative. The first immediate consequence of this is that trying to compute the logarithm in (15) might fail. Secondly, the optimization update (18) will try to maximize the left hand side of (26), which is an approximation of the likelihood function.

However, since the empirical moment estimates (9) will converge to the true moments (under suitable assumptions on the HMM) and, in turn, the estimates of the OOR parameters to their true values (by the continuous mapping theorem), we expect both of these issues to disappear as more data is made available. That HMMs estimated with method of moments using finite data can yield negative probability estimates is a known problem. Several workarounds have been proposed: truncation [26], projecting the estimates back to the feasible set using exterior point methods [31], etc. A full analysis of these issues and convergence is beyond the scope of this paper.

V. Numerical Evaluation

In this section we evaluate the performance of the proposed procedure on numerical examples using the following measures of accuracy. Firstly, the OOR parametrization θ enables estimation of posteriors over observations using (7) and (8). We define the prediction accuracy as the *root-mean-squared-error* (RMSE) of the output posterior vector (over y_k given $y_{1:k-1}$) compared to that of a standard HMM filter running with the true parameters.

Secondly, it can be shown [8, Lemma 3] that the OOR satisfies $\mathcal{B}_y = (BU)^T P^T \operatorname{diag}(Be_y)(BU)^{-T}$, so that

$$\mathcal{B} \overset{\text{def.}}{=} \sum_{y=1}^{Y} \mathcal{B}_y = (BU)^T P^T (BU)^{-T}. \qquad (27)$$

In other words, $\mathcal{B}$ is a similarly transformed version of the transition matrix P. Hence, a second natural choice of accuracy is the RMSEs of the eigenvalues of $\hat{\mathcal{B}}$ compared to those of P.

A. Simulations

Consider a fixed HMM with $X = Y = 3$. This system has 12 unknown parameters (in the standard HMM parametrization). Fig. 1 presents the RMSEs, for increasing data lengths, of the three eigenvalues of $\hat{\mathcal{B}}$ compared to those of the true transition matrix averaged over 100 realizations. Also plotted are the RMSEs of the estimated eigenvalues of P using EM started in the true parameters. The matrix $\hat{\mathcal{B}}$ was computed using spectral learning (SL); followed by one (SL^{+1}) and three (SL^{+3}) subsequent iterations of the proposed algorithm.

It can be seen from Fig. 1 (by the distances between the EM and SL^{+1} curves) that the statistical efficiency of the estimates is increased by one iteration of the proposed method. For smaller data sizes, there is a small gain in performing a few subsequent iterations – due to the quadratic approximation being worse when the initial estimate is further from the maximum. By the Perron-Frobenius theorem, the largest eigenvalue (λ_3) of P is placed in 1. The stationarity assumption in the proposed method enforces this, whereas it is apparent from Fig. 1 that SL estimates it.

Next, we consider randomly generated HMMs. In Fig. 2, the full distribution of the output posterior errors for fixed data sizes $N_{\text{train}} = 25\,000$ and $50\,000$ are plotted for 400 realizations with SL on the horizontal axis, and three iterations of the proposed method on the vertical axis. The dashed line shows the locii of points for which both methods give equal RMSE. The calculated likelihood was positive in 374 and 383, respectively, out of the 400 realizations. A majority of the points fall on the side corresponding to the proposed method having a better prediction accuracy. The points falling on the other side of the bisection can in part be explained by, as discussed in Section IV-C, that for finite data, we are only approximately maximizing the likelihood.

VI. Conclusions

This paper has considered identification of HMMs for the purpose of providing estimates of joint and posterior probabilities over observation sequences. A recently proposed method of moments from the machine learning community was extended by, as a second step, combining it with likelihood maximization to increase the statistical efficiency. In particular, since only substrings of length three (of the observed data) were used in the training procedure, some of the information available in correlations was lost. It was shown in simulations that a small number of maximization steps of a quadratic approximation of the likelihood was sufficient to get close to the performance of ML in terms of statistical accuracy.

References

[1] R. Durbin, ed., *Biological sequence analysis: Probabalistic models of proteins and nucleic acids*. Cambridge, UK: Cambridge University Press, 1998.

[2] M. Vidyasagar, *Hidden Markov Processes: Theory and Applications to Biology*. Princeton, NJ: Princeton University Press, 2014.

[3] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[4] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge, UK: Cambridge University Press, 2016.

[5] R. S. Mamon and R. J. Elliott, eds., *Hidden Markov Models in Finance*. Boston, MA: Springer US, 2014.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1993.

[8] D. Hsu, S. M. Kakade, and T. Zhang, "A spectral algorithm for learning hidden Markov models," *Journal of Computer and System Sciences*, vol. 78, pp. 1460–1480, Sept. 2012.

[9] A. Anandkumar, D. Hsu, and S. M. Kakade, "A method of moments for mixture models and hidden Markov models," in *Proceedings of the 25th Conference on Learning Theory (COLT'12)*, pp. 33.1–33.34, 2012.

[10] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2773–2832, 2014.

[11] B. Lakshminarayanan and R. Raich, "Non-negative matrix factorization for parameter estimation in hidden Markov models," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP'10)*, pp. 89–94, 2010.

[12] A. Kontorovich, B. Nadler, and R. Weiss, "On learning parametric-output HMMs," in *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, vol. 28, pp. 702–710, 2013.

[13] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Asymptotically efficient identification of known-sensor hidden Markov models," *arXiv:1702.00155 [cs.SY]*, 2017.

[14] S. M. Siddiqi, B. Boots, and G. J. Gordon, "Reduced-rank hidden Markov models," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, pp. 741–748, 2010.

[15] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola, "Hilbert space embeddings of hidden Markov models," in *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pp. 991–998, 2010.

[16] H. Jaeger, "Observable operator models for discrete stochastic time series," *Neural Computation*, vol. 12, no. 6, pp. 1371–1398, 2000.

[17] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. New York, NY: Springer, 2005.

[18] T. Söderström, P. Stoica, and B. Friedlander, "An indirect prediction error method for system identification," *Automatica*, vol. 27, no. 1, pp. 183 – 188, 1991.

[19] P. Van Overschee and B. L. De Moor, *Subspace identification for linear systems: Theory–Implementation–Applications*. Springer Science & Business Media, 2012.

[20] H. Hjalmarsson and B. Ninness, "Fast, non-iterative estimation of hidden markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, vol. 4, pp. 2253–2256, 1998.

[21] B. Vanluyten, J. C. Willems, and B. De Moor, "A new approach for the identification of hidden Markov models," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC'07)*, pp. 4901–4905, 2007.

[22] B. Vanluyten, J. C. Willems, and B. De Moor, "Structured nonnegative matrix factorization with applications to hidden Markov realization and clustering," *Linear Algebra and its Applications*, vol. 429, pp. 1409–1424, Oct. 2008.

[23] S. Andersson and T. Rydén, "Subspace estimation and prediction methods for hidden Markov models," *The Annals of Statistics*, vol. 37, no. 6B, pp. 4131–4152, 2009.

[24] L. Finesso, A. Grassi, and P. Spreij, "Approximation of stationary processes by hidden Markov models," *Mathematics of Control, Signals, and Systems*, vol. 22, pp. 1–22, Sept. 2010.

[25] G. Cybenko and V. Crespi, "Learning hidden Markov models using nonnegative matrix factorization," *IEEE Transactions on Information Theory*, vol. 57, pp. 3963–3970, June 2011.

[26] H. Jaeger, M. Zhao, K. Kretzschmar, T. Oberstein, D. Popovici, and A. Kolling, "Learning observable operator models via the ES algorithm," *New directions in statistical signal processing: From systems to brains*, 2005.

[27] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing," *Journal of Machine Learning Research*, vol. 17, no. 102, pp. 1–44, 2016.

[28] B. Balle, W. L. Hamilton, and J. Pineau, "Methods of moments for learning stochastic languages: Unified presentation and empirical comparison," in *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*, pp. 1386–1394, 2014.

[29] D. Tran, M. Kim, and F. Doshi-Velez, "Spectral M-estimation with application to hidden Markov models," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS'16)*, 2016.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[31] A. Shaban, M. Farajtabar, B. Xie, L. Song, and B. Boots, "Learning latent variable models by improving spectral solutions with exterior point methods," in *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI'15)*, pp. 792–801, 2015.

[32] A. Zaman, "Stationarity on finite strings and shift register sequences," *The Annals of Probability*, pp. 678–684, 1983.

[33] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *Journal of Global Optimization*, vol. 1, no. 1, pp. 15–22, 1991.

[34] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Philadelphia, PA: SIAM, 2nd ed., 2008.

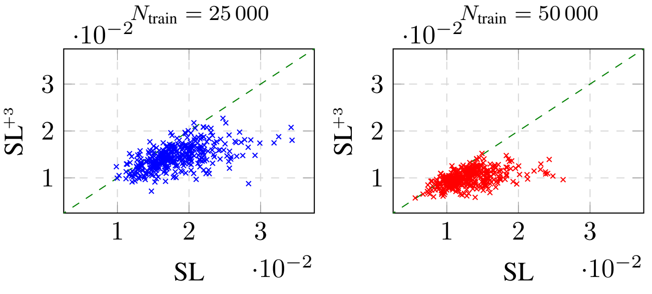

Fig. 2: Posterior errors (lower is better) of spectral learning (SL) and three subsequent iterations of the proposed procedure (SL^{+3}) for two different training data sizes. The dashed line shows the locii of points for which both methods give equal RMSE.