Variational Inference and Learning of Piecewise-linear Dynamical Systems
Xavier Alameda-Pineda, Vincent Drouard, Radu Horaud

To cite this version:
Xavier Alameda-Pineda, Vincent Drouard, Radu Horaud. Variational Inference and Learning of Piecewise-linear Dynamical Systems. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33 (8), pp.3753 - 3764. 10.1109/TNNLS.2021.3054407. hal-02745527v3

HAL Id: hal-02745527
https://hal.science/hal-02745527v3
Submitted on 26 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—Modeling the temporal behavior of data is of primordial importance in many scientific and engineering fields. Baseline methods assume that both the dynamic and observation equations follow linear-Gaussian models. However, there are many real-world processes that cannot be characterized by a single linear behavior. Alternatively, it is possible to consider a piecewise-linear model which, combined with a switching mechanism, is well suited when several modes of behavior are needed. Nevertheless, switching dynamical systems are intractable because their computational complexity increases exponentially with time. In this paper, we propose a variational approximation of piecewise linear dynamical systems. We provide full details of the derivation of two variational expectation-maximization algorithms, a filter and a smoother. We show that the model parameters can be split into two sets, static and dynamic parameters, and that the former parameters can be estimated off-line together with the number of linear modes, or the number of states of the switching variable. We apply the proposed method to the head-pose tracking, and we thoroughly compare our algorithms with several state of the art trackers.

Index Terms—Switching state space model, linear dynamical system, inverse regression, Bayesian inference, variational approximation, expectation-maximization, Kalman filter.

I. INTRODUCTION

Modeling the temporal behavior of data is of primordial importance in many scientific fields, such as signal processing [1], computer vision [2], [3], robotics [4], autonomous navigation [5], to cite just a few. The baseline model for addressing this problem is the linear dynamical system (LDS). The basic idea of LDS is to assume that both the dynamic and the observation equations follow linear-Gaussian models. This yields tractable learning and inference procedures, namely the Kalman filter (KF) [6] and the Kalman smoother (KS) [7].

In many situations, the latent (state) space, whose dynamics must be modeled, is embedded in a high-dimensional observed space. In general, the direct mapping, from the low-dimensional latent space to the high-dimensional feature (observed) space, as well as the inverse of this mapping, are non-linear. Moreover, the dynamics of the latent space may be non-linear as well. Several methods were proposed to deal with non-linear dynamical systems, e.g. Bayesian tracking with particle filters [8], [9], the extended Kalman filter (EKF) [10], and the unscented Kalman filter (UKF) [11]. Alternatively, it is possible to consider several linear dynamic models and to combine them with a switching mechanism that selects over time one among several linear regimes: this is referred to as the switching Kalman filter (SKF), the switching LDS, the jump-Markov linear system, or the switching state space model [12].

The mixture Kalman filter (MKF) [8] uses a sequential Monte Carlo method based on a random mixture of Gaussians to approximate the target distribution. It formulates non-linear systems into conditional or partial conditional LDSs. Outcomes of non-linear/non-Gaussian Bayesian trackers based on sequential importance sampling are reviewed and discussed in [9], most notably the problems of degeneracy, choice of importance density, and resampling. The basic idea of EKF is to linearize the equations using a first-order Taylor expansion and to apply the standard KF to the linearized model. The additional error due to linearization is not taken into account which may lead to sub-optimal performance. Rather than approximating a non-linear dynamical system with a linear one, UKF specifies the state distribution using a minimal set of deterministically selected sample points. The sample points, when propagated through the true non-linear system, capture the posterior state distribution accurately to the third order Taylor expansion. The stability of UKF was thoroughly investigated in the control literature, e.g. [13]. It was shown that the design of the noise covariance, of both state and observation equations, critically affects the performance of the filter.

The methods just outlined generally deal with a single non-linear or linearized, state equation. There are many real-world processes that cannot be characterized with a single state equation, but with multiple discrete modes of behavior, both in terms of their dynamics and of their observation model, in particular when the latter must predict (generate) high-dimensional observations from a low-dimensional state space.

We consider the problem of tracking the orientation of a person head/face (three rotation angles) from a sequence of images, referred to as head-pose tracking (HPT). A face detector provides input to a face descriptor immune to illumination changes, background conditions, as well as inter- and intra-person variabilities (shape and aspect). Face descriptors of choice are histograms of oriented gradients (HOGs) [14] and embeddings based on convolutional neural networks (CNNs) [15]–[18]. These high-dimensional feature vectors contain head pose information implicitly and a number of non-linear or piecewise-linear regression methods have been proposed to extract head pose, namely Gaussian process regression [19], support vector regression [20], kernel partial least squares [21],...
deep inverse regression [18], or Gaussian mixture of linear regressions [22], [23].

Recently it was proposed to approximate non-linear high-dimensional to low-dimensional (high-to-low) mappings with mixtures of linear-Gaussian [22], [24] and linear-Student [25] regressions. These models adopt an inverse regression strategy, namely they learn a low-to-high mapping followed by the evaluation of a high-to-low mapping. The rationale of this way of doing is manifold: (i) low-to-high regression learning avoids the estimation of a large number of parameters, hence it requires a relatively small amount of training data, (ii) the parameters of the high-to-low regression are analytically evaluated from the low-to-high parameters, (iii) the mixture model setting has the advantage of providing inference procedures using closed-form EM algorithms. It is interesting to note that these Gaussian/Student mixtures group data with similar regression associations together. Within the same cluster, the association can be considered as locally linear, which can then be resolved under the classical linear regression setting. This piecewise linear models are well suited to capture potentially non-linear relations. This was extensively discussed in [22] and in [24], and was successfully applied to both head-pose estimation [23] and audio-source localization [26], [27].

In this paper we propose a variational expectation-maximization algorithm to learn piecewise-linear dynamic systems (P-LDSs). A P-LDS can be viewed either as a piecewise-linear approximation of a non-linear dynamic system [9], or as a dynamic generalization of mixtures of linear regressions [22], [23], [26]. A P-LDS may also be viewed as soft version of switching LDS [12], [28]. The assignment variable of the piecewise-linear mixture model plays the role of the switching variable of both the dynamic and observation models and it is governed by the transition matrix of a corresponding hidden Markov model (HMM). It is well known that the complexity of these hybrid dynamical systems, i.e. systems that combine discrete- and continuous-valued latent variables, increases exponentially with time [12]. Indeed, for $K$ linear models and after $T$ time steps, the exact marginalized posterior distribution of the state is a Gaussian mixture with $K^2$ components. Therefore, the problem of learning the parameters of such hybrid systems must be carried out via approximate solutions. Traditionally, inference of hybrid dynamical systems is based on approximating the posterior distribution with a simpler one, e.g. the generalized pseudo Bayes filter. In this paper we propose an alternative based on replacing the difficult-to-compute posterior with an approximate tractable posterior.

The remainder of this paper is organized as follows. Section II describes related work. Section III formulates P-LDS and analyses its intractability. Section IV describes in detail the proposed variational approximation model as well as the as two EM algorithms, a variational filter and a variational smoother. For the sake of completeness, Section V describes a GPB2 approximation of P-LDS. Section VI describes experimental results obtained with head-pose tracking. 

\[^1\]Supplemental materials can be found at https://team.inria.fr/perception/research/learning-plds/.

II. RELATED WORK

The intractability of switching LDS (and of P-LDS) can be addressed using sampling methods: sequential Monte Carlo methods (particle filtering) have been used for this purpose. The main drawback is that sampling can be inefficient, which leads to slow convergence. To reduce the size of the state space, Rao-Blackwellisation may be employed. Instead of drawing the samples from the joint posterior of the discrete and continuous states, tractable sub-structures of the model can be utilized. Non parametric Bayesian inference of switching LDS was proposed in [29], where a switching mechanism is used for the dynamic model, while the observation model uses a standard LDS. This is problematic when the observation model is not linear. Moreover, a Gibbs sampler is used for inference, providing asymptotic properties tied to a high computational cost.

A general theory of Rao-Blackwellised particle filters (RBPFs) applied to dynamic Bayesian networks (DBNs) was proposed in [30]. In the case of switching LDS, marginalization of the joint posterior, namely analytic integration over the continuous variables, considerably reduces the sampling space. RBPF using Gibbs sampling was used for speech recognition [31], where the discrete state corresponds to phonemes and the continuous state corresponds to a time evolving representation of the observations. RBPF using Metropolis-Hastings sampling was used in [32] to analyse motion patterns of bees.

In [32] it was noticed that a naive exploration of the space of discrete variables is prohibitive and that data-driven (DD) MCMC sampling improves convergence. DD-MCMC requires supervised learning from a labeled training dataset, which may be cumbersome, if not prohibitive, because it is not practical to manually associate discrete-variable values with the observed vectors. More recently, [33] proposed a mixture of switching LDSs to analyse the dynamic behavior of pedestrians: an MCMC inference scheme uses both Gibbs and Metropolis-Hastings samplers. While potentially powerful, MCMC methods and their variants are non-analytic methods and typically suffer from slow convergence rates (they are only exact in the case of infinite size samples), especially in high-dimensional spaces, which is impractical in the case of tracking. A recent comparison between MCMC and variational inference emphasizes that the latter easily takes advantage of stochastic and distributed optimization [34].

The generalized pseudo Bayes (GPB) [35] and the GPB of order two (GPB2) [36], [37] algorithms belong to the assumed density filtering (ADF) [38] class of models which is widely used to approximate an intractable distribution with a simpler one. GPB collapses the mixture of $K$ Gaussian components, resulting from considering all the possible linear models at each time step, into one Gaussian component. GPB2 is more sophisticated and more time consuming than GPB, as it collapses the $K^2$ Gaussian components, resulting from considering all the possible linear models when going from one time step to the next, into $K$ Gaussians components. The GPB2 algorithm was applied to the analysis of motor cortical activity of hand movements in macaque monkeys.
i.e. the joint posterior distribution \( P \) is easier to compute. Variational models modify the structure of the posterior by removing dependencies between variables, i.e. the joint posterior distribution \( P \) is approximated by a tractable \textit{variational} distribution \( Q \) with variational parameters \( \theta \). The Kullback-Leibler divergence between \( Q \) and \( P \) is minimized with respect to \( \theta \). In the case of switching LDSs, [42], [28] and [43] propose to remove some of the dependencies between the continuous and discrete latent variables, thus yielding tractable solvers. The mixed-state DBN proposed in [42] is an HMM driving the LDS bias. In [28] an HMM switches between several LDSs, each LDS having a different latent variable with its own dynamic regime. Both [42] and [28] lead to an EM algorithm whose maximization step (learning) satisfies a set of fixed-point equations in the variational parameters. The variational model proposed in [43] is more general than the one proposed in [42] and in [28]. Nevertheless, their approximation breaks the dependencies between the HMM and the LDS as well as the temporal dependencies.

The variational model of [43] was applied to speech recognition [44] and to speech production [45], while the model of [42] was applied to human motion capture [46]. It is interesting to note that in spite of the recent popularity gained by variational models, e.g. [47], as they provide tractable solutions to various intractable inference problems, e.g. [2], [3], [48]–[56], at the best of our knowledge, variational inference of switching LDS has not been addressed for the last decade.

More recently, learning and inference of dynamical systems have been addressed in the framework of deep generative models (DGMs), where the linear-Gaussian transition and emission distributions of LDS are replaced with non-linear Gaussian models. In detail, the mean and covariance of a Gaussian distribution are modeled with neural networks. Because of this non-linear dependencies, direct optimization of the corresponding data log-likelihood function is intractable. This issue is solved by maximization of a variational lower bound of the log-likelihood. For example, [57] uses a recurrent neural network (RNN) to model the mean and diagonal covariance matrix. The proposed structured inference network corresponds to a deep Kalman smoother, that needs both past and future observations. This formulation belongs to a wider class of non-linear Gaussian state-space models that were recently reviewed in [58]. Deep neural networks can also be used within structured variational inference for pixel-level prediction tasks [59], [60], but we are not aware of any works addressing switching LDS with this methodology.

With respect to the related work just outlined, this paper has the following contributions. We propose a variational approximation of P-LDS. Unlike the variational model of [28], which switches between several linear regimes, we propose a piecewise-linear model that approximates non-linear dynamic models. Unlike [43], the proposed variational approximation doesn’t break the temporal dependencies. Unlike the RNN-based non-linear state-space model of [57], the proposed variational piecewise-linear model yields a closed-form solver and it has fewer parameters.

We provide full details of an EM algorithm that can be indifferentely used either as a \textit{variational filter} or as a \textit{variational smoother}. Unlike the models of [42] and of [28] that lead to solving a set of fixed-point equations, we develop a closed-form solution for learning the model parameters. Moreover, the proposed method benefits from a closed-form EM algorithm for the off-line estimation of the static parameters, namely, those associated with the observation model. This has two practical outcomes: (i) it reduces the task of learning to the estimation of the parameters of the dynamic model and (ii) it allows to learn the number of states of the switching variable (or, equivalently, the number of linear models) based on a model selection principle, i.e. the Bayes information criterion, [22], [25]. For the sake of completeness, we describe a GPB2 algorithm for P-LDS that slightly differs from the standard GPB2 for switching LDS in that it only needs to estimate the parameters of the dynamic model.

\section{Problem Formulation}

Let \( X \in \mathcal{X} \subset \mathbb{R}^L \) be a latent (or state) random variable and \( Y \in \mathcal{Y} \subset \mathbb{R}^D \) be an observation variable. Without loss of generality it will be assumed that the dimensionality of the observation space is much higher than the dimensionality of the latent space, \( D \gg L \). Let \( x \) and \( y \) denote realizations of \( X \) and \( Y \), respectively. Let \( t \in \mathbb{N} \) be the discrete time index: \( X_t \) denotes the latent variable at \( t \) and the notation \( X_{1:t} \) is a shorthand for the temporal sequence \( X_1, \ldots, X_t \). In an LDS, the observed vectors are connected to the latent vectors through an observation equation. We will consider the following piecewise-linear observation model. It is assumed that at any time \( t \) a realization \( (y_t, x_t) \) of \( (Y_t, X_t) \) is such that \( y_t \) is generated from \( x_t \) by a linear function \( y_t = l_k(x_t) \) plus an error term. At each time step \( t \), a discrete latent variable \( Z_t \) is introduced, such that \( Z_t = k \) if and only if \( y_t \) is the image of \( x_t \) by \( l_k \), with \( k \in \{1, \ldots, K\} \subset \mathbb{N} \). The \textit{piecewise} linear function that maps the state \( X_t \) onto the observation \( Y_t \) is:

\begin{equation}
    y_t = \sum_{k=1}^{K} \mathbb{I}(Z_t = k)(A_{Z_t} x_t + b_{Z_t} + e_{Z_t}),
\end{equation}

where matrix \( A_{Z_t=k} \in \mathbb{R}^{D \times L} \) and vector \( b_{Z_t=k} \in \mathbb{R}^D \) define \( l_k \), \( \mathbb{I}(\cdot) \) is the indicator function and \( e \sim \mathcal{N}(0, \Sigma) \) is a zero-mean Gaussian noise vector with covariance matrix \( \Sigma \in \mathbb{R}^{D \times D} \). The description of the model is completed by a similar piecewise linear dynamic equation:

\begin{equation}
    x_t = \sum_{k=1}^{K} \mathbb{I}(Z_t = k)(C_{Z_t} x_{t-1} + w_{Z_t}),
\end{equation}
where \( C_{\text{Z}_t} \in \mathbb{R}^{L \times L} \) is the state transition matrix and \( w \sim \mathcal{N}(0, \mathbf{Q}) \) is a zero-mean Gaussian noise vector with covariance matrix \( \mathbf{Q} \in \mathbb{R}^{L \times L} \). To summarize, the \( k \)-th LDS is defined by the following probabilistic model, see Fig. 1:

\[
\begin{align*}
    p(x_t | x_{t-1}, \text{Z}_t = k) &= \mathcal{N}(x_t; C_{\text{Z}_t} x_{t-1}, Q_{\text{Z}_t}), \\
    p(y_t | x_t, \text{Z}_t = k) &= \mathcal{N}(y_t; A_{\text{Z}_t} x_t + b_{\text{Z}_t}, \Sigma_{\text{Z}_t}), \\
    p(x_1) &= \mathcal{N}(x_1; \gamma_{\text{Z}_1}, \Gamma_{\text{Z}_1}), \\
    p(Z_1 = k) &= \pi_{\text{Z}_1},
\end{align*}
\]

where \( \{\gamma_k, \Gamma_k, \pi_k\}_{k=1}^K \) is a Gaussian mixture of parameters. Let, for example, \( D_{\text{Z}_t} \) denote vertical concatenation of \( D \) Gaussians.

Fig. 1. Graphical representation of the switching linear dynamic model that show the dependencies between the latent variables (white circles) and the observed variables (gray circles).

A. Computational Intractability

Exact model inference, i.e. estimation of the dynamic parameters \( \phi \), is faced with an intractability problem. Indeed, let’s analyze the complexity of computing the posterior distribution, namely the conditional probability of the state variable \( x_t \) given the present and past observations \( p(x_t | y_{1:t}) \). This distribution can be obtained by marginalization over the continuous and discrete variables given the observations:

\[
p(x_t | y_{1:t}) = \sum_{z_{1:t-1}} p(x_t, x_{1:t-1}, z_{1:t} | y_{1:t}) dx_{1:t-1},
\]

where \( \sum_{z_{1:t-1}} \) and \( dx_{1:t-1} \) are shorthands for \( \sum_{z_{1:t-1}} ^ dx_{1:t-1} \) and for \( dx_{1} \ldots dx_{t-1} \), respectively. Using the first-order Markov dependencies shown in the graphical model of Fig. 1, the joint probability (right hand side of (10)) can be factorized as:

\[
p(x_{1:t}, z_{1:t} | y_{1:t}) \propto p(x_1, z_1 | y_1) \times \prod_{t'=2}^{t} p(y_{t'}, x_{t'}, z_{t'}) p(x_{t'} | x_{t'-1}, z_{t'}) p(z_{t'} | z_{t'-1}).
\]

Substituting the factors of (11) with their expressions (3)-(7) and using standard properties of Gaussian distributions and using marginalization, the joint distribution (11) writes:

\[
p(x_t, x_{1:t-1}, z_{1:t} | y_{1:t}) = \beta_t \mathcal{N}([x_1; x_t]; \kappa_t, K_t),
\]

where the notation \( [x_1; x_t] \) denotes vertical concatenation of vectors \( x_1, \ldots, x_t \), and where the weight \( \beta_t \), mean \( \kappa_t \), and covariance \( K_t \) depend on the model parameters (8) and (9) and on \( Z_{1:t} \). Therefore, the predictive distribution (10) is a Gaussian mixture with a number of components that increases exponentially with time, i.e. there are \( K^t \) components after \( t \) time steps, which is intractable.

This phenomenon appears not only when attempting to evaluate \( p(x_{1:t} | y_{1:t}) \) (filtering), but as well as when \( p(x_{1:T} | y_{1:T}) \) (smoothing) is evaluated. While the former is used for on-line
prediction, i.e. when the model is already trained, the latter is part of the E-step of any algorithm used for learning the parameters, and therefore equally essential. In the following, we present our variational approximation to perform inference as well as the complete VEM algorithm for learning. We also discuss the derivation of the GPB2 algorithm [36] in the context of the proposed formulation.

IV. VARIATIONAL INFERENCE AND LEARNING

In this section we present a variational approximation of P-LDS and we derive a variational EM algorithm with tractable inference (expectation) and closed-form parameter learning (maximization). We assume that the continuous and discrete variables are independent, a posteriori: consequently, the joint distribution over \( x_{1:T} \) and \( z_{1:T} \) is approximated with the following factorization:

\[
p(x_{1:T}, z_{1:T} | y_{1:T}) \approx q(x_{1:T}) q(z_{1:T}).
\]

(13)

This follows the same philosophy as the factorial hidden Markov models [61]. However, here we deal with hybrid states, namely discrete and continuous, therefore the derivation [61] does not apply and we need to derive a new EM algorithm. Notice that the proposed model is different than the model of [28]. Indeed, the latter switches between several continuous states, with their own linear dynamic regimes, while the proposed model approximates a possibly non-linear dynamic regime with a piecewise-linear model, similar to GPB2, i.e. (44).

As with HMM and LDS learning, we assume that the entire sequence of observations is available for training and the challenge consists of inferring the entire chain of state variables and of estimating the model parameters, namely the parameter vectors \( \theta \), i.e. (8) (observation model) and \( \phi \), i.e. (9) (dynamic model). As already outlined in Section III, the parameters \( \theta \) don’t depend on time and they can be estimated using the algorithm of [22]. Therefore, we only need to estimate the dynamic parameters \( \phi \). Formally, we maximize the expected complete-data log-likelihood:

\[
Q(\phi) = \mathbb{E}_{p(x_{1:T}, z_{1:T} | y_{1:T})} \left[ \log p(x_{1:T}, z_{1:T}, y_{1:T} | \phi) \right],
\]

(14)

where the posterior distribution \( p(x_{1:T}, z_{1:T} | y_{1:T}) \) is evaluated with the model parameters at the previous iteration \( \phi^{\text{old}} \), implicit in the previous equation to simplify the reading.

A. Inference: The Expectation Steps

The two posterior distributions (13) write:

\[
\begin{align*}
\log q(z_{1:T}) &= \mathbb{E}_{q(x_{1:T})} \left[ \log p(x_{1:T}, z_{1:T} | y_{1:T}) \right] + \text{const}, \\
\log q(x_{1:T}) &= \mathbb{E}_{q(z_{1:T})} \left[ \log p(x_{1:T}, z_{1:T} | y_{1:T}) \right] + \text{const},
\end{align*}
\]

(15)

(16)

These distributions are alternatively evaluated, as explained in detail below.

1) E-Z step: By developing (15), ignoring the constant terms and using (3)-(7) we obtain:

\[
q(z_{1:T}) \propto \mathcal{N}(y_{1:T}; \eta_1 + b_{y_1}, \Sigma_{y_1}) \\
\times \exp \left( -\frac{1}{2} \text{tr} \left( \Sigma_{z_1}^{-1} A_{z_1} V_1 \right) \right) \\
\times \mathcal{N}(\eta_1; \gamma_{z_1}, \Gamma_{z_1}) \exp \left( -\frac{1}{2} \text{tr} \left( \Gamma_{z_1}^{-1} V_1 \right) \right) \\
\times \prod_{t \geq 2} \left[ \mathcal{N}(y_{1:T}; A_{z_1} \eta_t + b_{z_1}, \Sigma_{z_1}) \\
\times \exp \left( -\frac{1}{2} \text{tr} \left( \Sigma_{z_1}^{-1} A_{z_1} V_t \right) \right) \\
\times \mathcal{N}(\eta_t; C_{z_1} \eta_{t-1}, Q_{z_1}) \\
\times \exp \left( -\frac{1}{2} \text{tr} \left( C_{z_1} Q_{z_1}^{-1} C_{z_1} V_{t-1} + Q_{z_1}^{-1} V_t \right) \right) \\
\times \exp \left( \text{tr} \left( Q_{z_1}^{-1} C_{z_1} W_t \right) \right) \right]_t \tau_{z_1-1z_1},
\]

(17)

where \( \eta_t = \mathbb{E}_{q(x_{1:T})} [x_{t}] \) is the posterior mean of \( x_t \), \( V_t = \mathbb{E}_{q(x_{1:T})} [x_{t} x_{t}] - \eta_t \eta_t^T \) is the posterior covariance of \( x_t \) and \( W_t = \mathbb{E}_{q(x_{1:T})} [x_{t-1} x_{t}] - \eta_{t-1} \eta_t^T \) is the posterior cross-covariance of \( x_{t-1} \) and \( x_t \).

One may notice that it is possible to group the terms that depend on \( z \) in (17), thus yielding:

\[
q(z_{1:T}) \propto \rho_{t,z_1}^1 \prod_{t \geq 2} \left( \rho_{t-1, t \tau_{z_1-1z_1}} \right),
\]

(18)

which is equivalent to an HMM with observation probabilities \( \rho_{t,z_1} \). Therefore, by computing the standard forward-backward algorithm for HMMs, one can easily obtain the forward \( \rho_{t,z_1}^f \) and backward \( \rho_{t,z_1}^b \) probabilities to eventually obtain the posterior probability \( q(z_{1:T} | y_{1:T}) \approx q(z_1) \):

\[
q(z_{1:T}) = \rho_{t,z_{t-1}z} \left( \frac{\rho_{t-1, t \tau_{z_1-1z_1}} \rho_{t,z_1}^b}{\sum_{z_{t-1}} \rho_{t-1, t \tau_{z_1-1z_1}} \rho_{t,z_1}^b} \right).
\]

(19)

The estimation of the transition parameters \( \tau_{z_{t-1}z_1} \) requires the joint posteriori probability distribution of \( z_{t-1}, z_t \), which can be easily computed as:

\[
q(z_{t-1}, z_t) = \rho_{t-1, t \tau_{z_1-1z_1}} \rho_{t,z_{t-1}z_1} \rho_{t,z_1}^b \rho_{t,z_{t-1}z_1}^b.
\]

(20)

2) E-X step: By developing (16), ignoring the constant terms and using (3)-(7) we obtain:

\[
\begin{align*}
\log q(x_{1:T}) &\propto \\
&-\frac{1}{2} \left( x_t^T (V_1)^{-1} x_t - 2 x_t^T (V_1)^{-1} \eta_t + x_t^T (\Gamma)^{-1} x_t - 2 x_t^T (\Gamma)^{-1} \eta_t \right) \\
&+ \sum_{t \geq 2} \left( x_t^T (V_1)^{-1} x_t - 2 x_t^T (V_1)^{-1} \eta_t + x_t^T (\Gamma)^{-1} x_t - 2 x_t^T (\Gamma)^{-1} \eta_t \right) \\
&- 2 x_t^T R_t x_t + 2 x_t^T \left( S_t \right)^{-1} x_t (1 - x_t^T (S_t)^{-1} x_t),
\end{align*}
\]

(21)
where the following definitions hold:

$$\eta_t^i = V_t^i \left( \sum_z \rho_{t,z} A_z^T \Sigma_z^{-1} (y_t - b_z) \right),$$  \quad (22)

$$(V_t^i)^{-1} = \sum_z \rho_{t,z} A_z^T \Sigma_z^{-1} A_z,$$  \quad (23)

$$(\Gamma_t)^{-1} = \sum_z \rho_{t,z} \Gamma_z^{-1},$$  \quad (24)

$$(\bar{Q}_t)^{-1} = \sum_z \rho_{t,z} Q_z^{-1},$$  \quad (25)

$$R_t = \sum_z \rho_{t,z} Q_z^{-1} C_z,$$  \quad (26)

$$(S_t)^{-1} = \sum_z \rho_{t,z} C_z^T Q_z^{-1} C_z,$$  \quad (28)

In order to estimate the parameters of the dynamics, $C_z$ and $Q_z$, one needs the joint posterior probability of $x_t$ and $x_{t-1}$:

$$q(x_t, x_{t-1}) = \mathcal{N}(x_t | x_{t-1}; \eta_t^i, V_t^i).$$  \quad (30)

To be valid, the last equation requires that $\rho_{t,z} C_z^T Q_z^{-1} C_z$ is invertible for all values of $z$, which implies that $C_z$ is a full rank matrix for all values of $z$, which is a very mild assumption. This is very close to a standard LDS (Kalman filter), but different enough in that standard forward-backward recursions cannot be applied. Indeed, in a standard LDS the following relationship holds: $R_t Q_t^{-1} R_t = S_t^{-1}$, which is not the case in general. This condition is equivalent to impose the same Gaussian dynamic model to all the realizations of $Z_t$ (which clearly corresponds to a Kalman filter). However, from (21), one may easily see that the joint posterior distribution is a Gaussian dynamic model to all the realizations of $Z_t$, which is a very mild assumption.

The forward initialization with $(V_t^1)^{-1} = \Gamma_t^{-1} + (V_t^1)^{-1}$ and $\eta_t^1 = V_t^1 \left( \Gamma_t^{-1} \eta_t^1 + (V_t^1)^{-1} \eta_t^1 \right)$. The backward recursion starts at $t = T - 1$ with $(V_t^b)^{-1} = \eta_t^b$ (and therefore the value of $\eta_T^b$ has no effect). Together, they allow us to write the posterior probability of $x_t$:

$$q(x_t | y_{1:T}) = \mathcal{N}(x_t | \eta_t, V_t),$$  \quad (29)

$$\eta_t = V_t \left( (V_t^b)^{-1} \eta_t^b + (V_t^1)^{-1} \eta_t^1 \right),$$

$$(V_t)^{-1} = (V_t^b)^{-1} + (V_t^1)^{-1}.$$  \quad (31)

The optimal transition parameters of the transition parameters correspond to a standard HMM model and are given by:

$$\tau_{zz'} = \frac{1}{T-1} \sum_{t=2}^T \rho_{t,zz'}. (34)$$

The optimal value of $Q_z$ is obtained by taking the derivative of $Q$ with respect to $Q_z$ and setting this derivative equal to zero:

$$Q_z = \frac{1}{\sum_{t \geq 2} \rho_{tz}} \sum_{t \geq 2} \rho_{tz} (\eta_t - C_z \eta_{t-1}) (\eta_t - C_z \eta_{t-1})^T + C_z V_{t-1} C_z^T + V_t - 2 C_z W_t. (35)$$
Taking the derivative of $Q$ with respect to $C_z$ is more involved since one needs to take the derivative of the matrix-trace operator. After setting the derivatives equal to zero we obtain:

$$C_z = \sum_{t \geq 2} \mu_t z (W_t^T + \eta_t \eta_{t-1}^T) \left( \sum_{t \geq 2} \mu_t z (V_{t-1} + \eta_{t-1} \eta_{t-1}^T) \right)^{-1}$$

(36)

V. GPB2 INFERENCE AND LEARNING

GPB2 [36] is a commonly used algorithm to deal with the intractability of switching LDSs. For the sake of completeness, we describe the GPB2 algorithm for the particular case of P-LDS. As above, only the dynamic parameters $\phi$, i.e. (9), need to be estimated. The central idea of GPB2 is to recursively collapse a $K^2$-component mixture into a $K$-component mixture, as explained below. This implies that at each time index $t$ the conditional posterior $p(x_t|y_{1:t})$ is successively approximated. To do that, the marginalization chain in (10) is truncated, yielding:

$$p(x_t|y_{1:t}) = \sum_{z_{t-1} = 1}^{K} \sum_{z_t = 1}^{K} \int p(x_{t-1}, x_t, z_{t-1}, z_t|y_{1:t}) dx_{t-1},$$

(37)

with:

$$p(x_{t-1}, x_t, z_{t-1}, z_t|y_{1:t}) \propto p(y_t|x_t, z_t)p(x_t|x_{t-1}, z_{t-1})p(z_t|z_{t-1})p(x_{t-1}, z_{t-1}|y_{1:t-1}).$$

(38)

GPB2 yields the following approximation:

$$p(x_{t-1}, z_{t-1}|y_{1:t-1}) = \tilde{\rho}_{t-1, z_{t-1}, z_t}^F \mathcal{N}(x_{t-1}|\tilde{\eta}_{t-1, z_{t-1}, z_t}^F, \tilde{V}_{t-1, z_{t-1}, z_t}^F),$$

(39)

where $\tilde{\rho}_{t-1, z_{t-1}, z_t}^F$, $\tilde{\eta}_{t-1, z_{t-1}, z_t}^F$ and $\tilde{V}_{t-1, z_{t-1}, z_t}^F$ play the same role as $\rho_{t-1, z_{t-1}, z_t}$, $\eta_{t-1, z_{t-1}, z_t}$ and $V_{t-1, z_{t-1}, z_t}$ in the previous section, but they take different numerical values because they are computed using the GPB2 approximation instead of the variational one.

By using (3), (4), (7) and (38), the approximation in (39) boils down to the following relationship:

$$p(x_t|y_{1:t}) \approx \sum_{z_{t-1} = 1}^{K} \sum_{z_t = 1}^{K} \tilde{\rho}_{t, z_{t-1}, z_t}^F \mathcal{N}(x_t|\tilde{\eta}_{t, z_{t-1}, z_t}^F, \tilde{V}_{t, z_{t-1}, z_t}^F),$$

(40)

where the priors $\tilde{\rho}_{t, z_{t-1}, z_t}^F$, the means $\tilde{\eta}_{t, z_{t-1}, z_t}^F$, and the covariances $\tilde{V}_{t, z_{t-1}, z_t}^F$ are given by:

$$\tilde{\rho}_{t, z_{t-1}, z_t}^F = \tilde{\rho}_{t-1, z_{t-1}, z_t}^F \mathcal{N} \left( d_{t, z_{t-1}, z_t}; 0, S_{t, z_{t-1}, z_t} \right),$$

(41)

$$\tilde{\eta}_{t, z_{t-1}, z_t}^F = \tilde{V}_{t, z_{t-1}, z_t} \left( A_{t, z_{t-1}} \Sigma_{z_{t-1}}^{-1} (y_t - b_{z_t}) + P_{t, z_{t-1}, z_t} C_{z_t} \tilde{\eta}_{t-1, z_t}^F \right),$$

(42)

$$\tilde{V}_{t, z_{t-1}, z_t}^F = \left( A_{t, z_{t-1}} \Sigma_{z_{t-1}}^{-1} A_{z_t} + P_{t, z_{t-1}, z_t} \right)^{-1},$$

(43)

with:

$$d_{t, z_{t-1}, z_t} = y_t - A_{z_t} (C_{z_t} \tilde{\eta}_{t-1, z_t}^F) - b_{z_t},$$

$$S_{t, z_{t-1}, z_t} = \Sigma_{z_t} + A_{z_t} (Q_{z_t} + C_{z_t} \tilde{V}_{t-1, z_t}^F C_{z_t}^T) A_{z_t}^T,$$

$$P_{t, z_{t-1}, z_t} = (Q_{z_t} + C_{z_t} \tilde{V}_{t-1, z_t}^F C_{z_t}^T)^{-1}.$$

Consequently, the dynamic model expands the $K$-component GMM hypothesized in (39) into a $K^2$-component GMM. GPB2 collapses this $K^2$ components into $K$ components by moment matching, thus obtaining:

$$p(x_t|y_{1:t}) \approx \sum_{z_t = 1}^{K} \tilde{\rho}_{t, z_t}^F \mathcal{N}(x_t|\tilde{\eta}_{t, z_t}^F, \tilde{V}_{t, z_t}^F),$$

(44)

with $\tilde{\rho}_{t, z_t}^F$, $\tilde{\eta}_{t, z_t}^F$, $\tilde{V}_{t, z_t}^F$ are given by:

$$\tilde{\rho}_{t, z_t}^F = \sum_{z_{t-1} = 1}^{K} \tilde{\rho}_{t, z_{t-1}, z_t}^F,$$

(45)

$$\tilde{\eta}_{t, z_t}^F = \sum_{z_{t-1} = 1}^{K} \frac{\tilde{\rho}_{t, z_{t-1}, z_t}^F}{\tilde{\rho}_{t, z_t}^F} \tilde{\eta}_{t, z_{t-1}, z_t}^F,$$

(46)

$$\tilde{V}_{t, z_t}^F = \sum_{z_{t-1} = 1}^{K} \frac{\tilde{\rho}_{t, z_{t-1}, z_t}^F}{\tilde{\rho}_{t, z_t}^F} \tilde{V}_{t, z_{t-1}, z_t}^F \left( \tilde{\eta}_{t, z_{t-1}, z_t}^F - \tilde{\eta}_{t, z_t}^F \right)^T.$$
all our experiments. It is worthwhile to notice that the offline training procedure is shared by the variational filter, the variational smoother and GPB2.

### A. Experimental Setup

We empirically evaluate the performance of the proposed methods with three publicly available datasets: Biwi-Kinect [62], EYEDIAP [63] and Vernissage [64] (see Table I):

- The Biwi-Kinect dataset contains 24 videos of 20 people (16 men and 4 women) recorded with a Kinect camera. During the recordings, people were asked to move their heads freely in front of the camera. 3D head pose (pitch, yaw, and roll angles) annotations were obtained automatically and accurately for each video frame using the face-shift software. The angle values range from $-60^\circ$ to $60^\circ$ for pitch, $-75^\circ$ to $75^\circ$ for yaw and $-20^\circ$ to $20^\circ$ for roll. The dataset provides RGB and depth images as well as the calibration matrices. The 3D nose positions are provided as well.

- The EYEDIAP dataset is intended for both eye-gaze and head-pose estimation. It contains 94 videos of 16 people recorded using different configurations, e.g. static and rotating heads. The dataset provides RGB videos, with both HD and VGA resolution, and depth videos with the associated calibration matrices. Annotations of both head-pose and eye-gaze are provided for each video frame. The angle values range from $-40^\circ$ to $40^\circ$ for pitch and $-50^\circ$ to $50^\circ$ for yaw.

- The Vernissage dataset contains 10 recordings of 20 people interacting with each and with a robot. Each recording comprises two people. The dataset was recorded with a camera embedded into the head of a robot head. A network of infrared cameras combined with optical markers placed on the participants’ heads provide accurate ground-truth head positions and orientations.

### B. Implementation details

Facial features are computed as follows. A face detector provides bounding boxes for each frame and for each video. Then a high-dimensional feature vector is extracted from each bounding box using histogram of oriented gradients (HOG) descriptors, obtained along the implementation described in [23] which yields vector-valued observations of dimension $D = 1888$. For all datasets and for each face identity, we split the corresponding videos into two disjoint sets: a training set and a test set. Since the datasets are annotated, i.e. there are ground-truth head-pose parameter values associated with each frame. As already mentioned, we use the method of [23] to estimate $\theta$.

The dynamic parameters are initialized as follows. First we set $C_k = I$ since we noticed that the simultaneous estimation of $C_k$ and $Q_k$ is subject to instabilities in the estimation of the dynamic parameters. The covariance matrices $\{C_j, Q_j\}_{j=1}^K$ are initialized with the identity matrix. The entries of the transition matrix $\{\tau_{ij}\}_{i,j=1}^K$ are initialized in the following way. We compute the pairwise Bhattacharyya distances [65] between the $K$ Gaussian components defined by (5). The variational EM algorithm alternates between inference of $q(Z_t)$ (E-Z step) and of $q(x_t)$ (E-X step). The variational parameters $\eta_t$ and $V_t$ are initialized with their previously estimated values, namely: $\eta_t = \eta_{t-1}$ and $V_t = V_{t-1}$.

We implemented both the variational filter and the variational smoother described in detail in the previous section. The main difference between the filter and the smoother is the amount of information that is used at inference time. Indeed, while the filter uses only causal information (i.e. past observations) the smoother uses also non-causal information (i.e. past and future observations). This is why, a priori, one expects the smoother to have better performance, at the price of being a completely off-line algorithm that cannot be used for real-time applications.

### C. Results and Discussion

The proposed methods were compared with the following state-of-the-art HPT methods:

- **Landmarks** [66], [67] combines 2D face landmark detection with head pose estimation and with tracking. Head pose is estimated in the following way: the 3D landmarks of a mean face are projected onto the image plane and the error between the projected landmarks and the observed landmarks is minimized over the pose parameters. At each time step this non-linear minimizer is initialized with the pose parameters computed at the previous time step. The publicly available implementation of this method only computes the pitch and yaw angles.

- **OpenFace** [68] is an open source software package for facial behavior analysis, i.e. facial landmark detection and tracking, head pose and eye gaze estimation. It extracts 68 2D facial landmarks using conditional local neural fields and tracks them over time with a three-layer CNN trained to predict landmark detection errors [69]. Once 2D landmarks are detected and tracked, they are used in conjunction with a 3D facial model to compute head pose parameters.

- **ICP tracking** [70] uses both depth and color (RGB-D images) from which 16 3D facial landmarks are manually

### TABLE I

**Summary of principal features of the datasets used for empirical evaluation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Recordings</td>
<td>24</td>
<td>94</td>
<td>10</td>
</tr>
<tr>
<td>#Participants</td>
<td>20</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>Sensor type</td>
<td>RGB-D</td>
<td>RGB-D</td>
<td>RGB</td>
</tr>
<tr>
<td>Pitch range</td>
<td>$[-60^\circ, +60^\circ]$</td>
<td>$[-40^\circ, +40^\circ]$</td>
<td>$[-90^\circ, +90^\circ]$</td>
</tr>
<tr>
<td>Yaw range</td>
<td>$[-75^\circ, +75^\circ]$</td>
<td>$[-50^\circ, +50^\circ]$</td>
<td>$[-90^\circ, +90^\circ]$</td>
</tr>
<tr>
<td>Roll range</td>
<td>$[-20^\circ, +20^\circ]$</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Annotation</td>
<td>Automatic fitting with a deformable 3D shape model</td>
<td>Optical motion capture device</td>
<td>N/A</td>
</tr>
</tbody>
</table>
The landmarks are tracked based on estimating the rigid motion between consecutive frames.

These HPT methods make use of 2D facial landmarks, whose detection is known to be sensitive to large head poses that induce partial occlusions of the face. This stays in contrast with the proposed method that directly exploits high-dimensional descriptors of faces. For completeness, we also compared our algorithms with an implementation of the Kalman filter (LDS), namely (1)-(6) with \( K = 1 \), with the HPE method of [23] and with GPB2, i.e. Section V and [71]. To quantitatively evaluate HPT we compute average and standard deviation of the absolute error between the estimated pose parameters and the ground-truth parameters provided with each annotated dataset.

The results obtained with the Biwi-Kinect, EYEDIAP and Vernissage datasets are summarized in tables II, III and IV. The HPT method of [66], [67] does not estimate the roll angle. Moreover, this method relies on 2D landmark detection. Therefore, it fails whenever all the landmarks are not properly detected. This is the reason for which this method did not provide publishable results, when applied to the EYEDIAP and Vernissage datasets. The method of [70] takes as input RGB-D images and manually annotated facial landmarks, hence it could only be applied to the EYEDIAP dataset. In general, the proposed descriptor-based trackers perform better than the landmark-based trackers. A lower standard deviation corresponds to a higher precision and a better method repeatability.

In order to assess the statistical differences in performance between the various methods, we used the Wilcoxon signed-rank test [72], inspired from [73], which is a non-parametric statistical hypothesis test that is commonly used to compare two related samples in order to assess the null hypothesis that the median difference between pairs of observations is zero. It can be used as an alternative to the paired Student’s t-test (t-test for matched pairs) when the population cannot be assumed to be normally distributed. For most of the comparisons, there were no statistical differences. The only notable exception is OpenFace [68], which performs considerably worse than the other methods on two of the three datasets: the yaw and roll estimation for the Biwi-Kinect dataset, and the pitch and yaw estimation for the EYEDIAP dataset. These estimation are marked with an asterisk in Table II and Table III, respectively.

As an example, Figure 3 shows a ground-truth yaw (left-
right rotation) trajectory as well as trajectories obtained with HPE_GLLiM [23], with HPT_GPB2 (top) and with the proposed variational filter (bottom). Overall, the performance of the proposed variational EM algorithms is comparable with the performance of GPB2. Notice that the large errors produced by HPE_GLLiM are filtered by both trackers.

We measured the CPU time needed to compute one time step. This comprises the following processes: (i) extraction of a HOG descriptor from a face, (ii) head pose estimation, and (iii) tracking, where the tracker can be either the GPB2 or the variational filter. Using an Intel-Xeon and Matlab, it takes 1.0 second to extract a HOG descriptor and to estimated head pose, 8.55 seconds to run GPB2 and 2.45 seconds to run the variational filter, respectively.

VII. CONCLUSIONS

In this paper we addressed the problem of learning and inference of piecewise LDS. The latter belongs to the switching LDS class of models, which is known to be intractable because of the combinatorial explosion, over time, of the modes of the latent space. The standard way of dealing with this problem is to use the GPB2 algorithm which collapses a $K^2$-component mixture into a $K$-component one based on moment matching – a computationally demanding process.

Alternatively, we propose a variational approximation of P-LDS and two associated EM algorithms: a variational filter and a variational smoother. Both the filter and the smoother are based on closed-form expressions, which guarantees efficient computation and fast convergence. The proposed variational filter is of the order of 3.5 times faster than the GPB2 method. Not surprisingly, the most time-consuming part of GPB2 is the moment matching – a computationally demanding process. Using an Intel-Xeon and Matlab, it takes 1.0 second to extract a HOG descriptor and to estimated head pose, 8.55 seconds to run GPB2 and 2.45 seconds to run the variational filter, respectively.

We applied the proposed algorithms to the problem of head-pose tracking. We presented a series of experiments using several datasets. We carried out a benchmark that included our algorithms and several state-of-the-art tracking algorithms. We note that the variational-based tracker compares well with GPB2 and with the other trackers. It should be noted, however,
that the landmark based methods, e.g. [66], [67], fail to track in many cases. The best performing method is [70]. Nevertheless, this method has limitations because it requires manual annotation of facial landmarks and it can only be applied to RGB-D images. In contrast, the proposed method is based on extracting high-dimensional descriptors from RGB images, and appears to be more robust against facial self-occlusions than the landmark-based methods.

REFERENCES


