# Hierarchical Visualization of Time-Series Data Using Switching Linear Dynamical Systems

**Onno Zoeter**[†] **Tom Heskes**

SNN, University of Nijmegen

Geert Grooteplein 21, 6525 EZ, Nijmegen, The Netherlands

{*orzoeter, tom*}*@snn.kun.nl*

[†]Corresponding author

**Abstract**

We propose a novel visualization algorithm for high-dimensional time-series data. In contrast to most visualization techniques we do *not* assume consecutive data points to be independent. The basic model is a linear dynamical system which can be seen as a dynamic extension of a probabilistic principal component model. A further extension to a particular *switching* linear dynamical system allows a representation of complex data onto multiple and even a hierarchy of plots. Using sensible approximations based on expectation propagation the projections can be performed in essentially the same order complexity as its static counterpart. We apply our method on a real-world data set with sensor readings from a paper machine.

*Keywords* – **data visualization, time-series, latent variables, principal component analysis, switching linear dynamical systems, approximate inference**

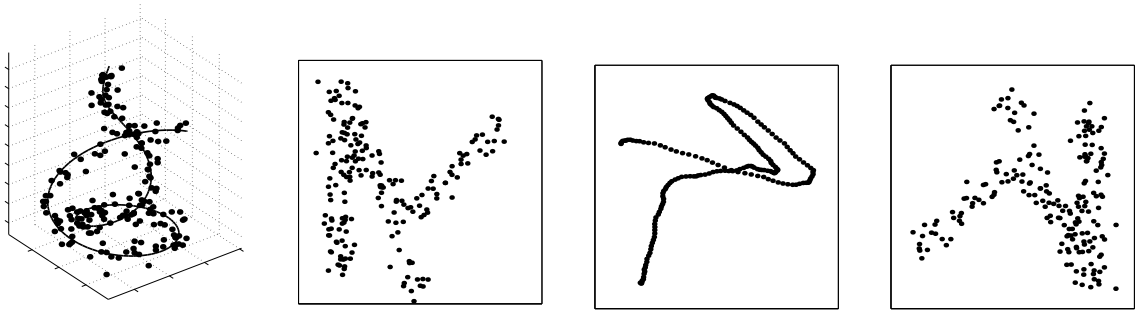
## I. INTRODUCTION

A data visualization problem can be interpreted as a learning and inference problem in a probabilistic generative model. In many settings, the goal of visualization is to show the user a projection of a high-dimensional dataset onto a lower-dimensional manifold, which often (and also in this article) is taken to be two-dimensional. The coordinates on this manifold can be treated as latent variables $\mathbf{x}$. The high-dimensional observations $\mathbf{y}$ are related to the latent variables through a probabilistic generative model $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ the parameters of the model. Given a set of observations $\mathbf{y}_{1:T}$, a projection can be made by plotting the posterior means $E[\mathbf{x}_t|\mathbf{y}_{1:T}, \boldsymbol{\theta}]$ for all $t$, where the average is over the posterior

$$p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \propto p(\mathbf{x}_{1:T}|\boldsymbol{\theta}) \prod_t p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}) \,,$$

with $p(\mathbf{x}_{1:T}|\boldsymbol{\theta})$ the prior distribution of the latent variables. An elegant example of this is the probabilistic principal component analysis (PPCA) model presented in [1], or the equivalent and independently proposed sensible principal component analysis model [2]. In this model all observations are assumed to be independent, which is reflected in a factorized prior for the latent variables. The projection is linear and both the prior and output noise model are spherical Gaussians:

$$\begin{aligned} p(\mathbf{x}_t|\boldsymbol{\theta}) &= \mathcal{N}(\mathbf{x}_t; \mathbf{0}, I) \\ p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t + \boldsymbol{\mu}, r^2 I) \,. \end{aligned}$$



(a) Data (b) PCA projection (c) LDS projection (d) LDS projection of permuted data

Fig. 1. PCA and LDS projections of toy data. Note that the scale and orientation of the projections are not identified and can be chosen arbitrarily. The same curve can be seen in all three projections.

In the above $\mathcal{N}(.;.,.)$ denotes the Gaussian probability density function. Since $C\mathbf{x}_t$ is a linear combination of the two columns of $C$, this model describes a 2D hyperplane in observation space. To take into account that we do not expect the data to lie exactly on this 2D hyperplane, Gaussian noise with covariance $r^2 I$ is added in the remaining directions. In [1] it is shown that such a model, with $\boldsymbol{\theta} = \{C, \boldsymbol{\mu}, r^2\}$ set to its maximum likelihood value, is functionally equivalent to the well-known principal component analysis model. The high-dimensional data will be projected onto the principal subspace, the columns of $C$ span this principal subspace, and $r^2$ can be interpreted as the variance that is lost by projecting.

Although (P)PCA is a useful method for many applications, it is suboptimal for time-series data, where the independence assumption is clearly violated. In this article we therefore use a linear dynamical system (LDS) also known as Kalman filter model for the visualization of time-series data. An LDS with spherical observation noise can be interpreted as "PPCA through time". In addition to the standard PPCA model, dependencies between observations are modeled by linear Gaussian transitions on the latent variables

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_{t+1}; A\mathbf{x}_t, Q)\,,$$

where $\boldsymbol{\theta}$ now also includes the matrices $A$ and $Q$. A suitable prior distribution for $\mathbf{x}_1$ is discussed in Appendix A.

The latent variables in the LDS model do not only provide a representation of the structure of the data within one time step, but also between time steps. Whereas in the static (PPCA) case the posterior factorizes: $E[\mathbf{x}_t|\mathbf{y}_{1:T}, \boldsymbol{\theta}] = E[\mathbf{x}_t|\mathbf{y}_t, \boldsymbol{\theta}]$, and the data points are projected independently, in the LDS case the posterior does not factorize and the entire data sequence will contribute to the projection of $\mathbf{y}_t$. The linear transition $A$ can be thought of as a parameter of Brownian motion in latent space. An $A$ matrix close to 0 indicates that the correlation between observations is very low (with $A = 0$ PPCA is retrieved): the projection of previous and future observations will give very little information about the projection of the current observation. When $A$ is close to $I$, successive points are expected to lie close together and the data trajectory is expected to be smooth.

A toy problem illustrates these intuitions. Figure 1(a) shows an artificial data set. The underlying true trajectory of the system is presented with a solid line, dots give noisy measurements. A PCA projection is given in Figure 1(b) revealing some of the structure, but still masking the relative smooth trajectory through time. In Figure 1(c) the same data is visualized using an LDS, showing clearly the smooth and relatively regular trajectory. Since the PPCA model assumes the data to be independent, a permuted version of the data set would result in an identical projection. For the LDS, which explicitly tries to model dependencies, we *do* obtain a different projection. The LDS projection of a random permutation of the data set in Figure 1(d) is hardly distinguishable from the PCA projection (as explained in Appendix A, since the scale and orientation of the projection cannot be identified from the data, we are allowed to rotate the figure 180° to see the similarities). The impact of a random permutation is also reflected in the maximum likelihood estimate of $A$, which is roughly $.99I$ for the original projection in Figure 1(c) and $.07I$ for the second one in Figure 1(d).

Many real-world phenomena show structures at different levels of detail. For instance in an industrial process clusters on the largest scale may correspond to normal production and failure modes. A single projection would only reveal these already very well known clusters. To make the proposed visualization method useful for exploratory purposes we extend it such that interactively sub-models can be fitted. In Section II, after discussing

the LDS projection in more detail, we extend the underlying model to a switching linear dynamical system (SLDS). This allows a projection onto multiple plots. In Section III this SLDS model is extended further such that sub-models can be fitted recursively. From the perspective of an end user this corresponds to zooming in on areas of interest. From a modeler's point of view it can be seen as adding extra degrees of freedom in regions where it is needed. This dynamical hierarchy is an extension of the model presented in [3]. We apply the model in Section V on a real-world data set with high-dimensional sensor readings of a running paper machine.

## II. VISUALIZATION WITH AN SLDS

### A. MAXIMUM LIKELIHOOD

Given a data set $\mathbf{y}_{1:T}$, we find the maximum likelihood settings of the parameters in the LDS

$$\boldsymbol{\theta}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathbf{y}_{1:T}|\boldsymbol{\theta}) \ ,$$

using an EM-algorithm. A local maximum[1] is found by alternating an Expectation step in which posteriors $p(\mathbf{x}_t|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\mathrm{old}})$ are inferred using the well-known Kalman smoother, and a Maximization step in which the expected complete data log-likelihood

$$\hat{\mathcal{L}}(\boldsymbol{\theta}) = E_{p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\mathrm{old}})} \left[\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta})\right] \ ,$$

is maximized w.r.t. $\boldsymbol{\theta}$. The standard EM-algorithm for an LDS is well known. It is a special case of the E-step outlined in Section IV and the M-step presented in Appendix C. It can easily account for missing observations. For initialization, we use the crude PPCA estimate proposed in [3] with $A = 0$.

To ensure the interpretability of the final projections, the EM-algorithm must be implemented with some care. Details are given in Appendix A.

### B. MULTIPLE PLOTS

In an LDS for visualization, we model the high-dimensional data with a single two-dimensional hyperplane, yielding a single projection. Here we extend the LDS to a *switch-*

[1] In the LDS model with the stationary distribution as prior one can show that there is only a single plateau of invariant parameter settings that have equal likelihood. When the model is extended in Section II-B the EM-algorithm can only be guaranteed to find a local maximum.

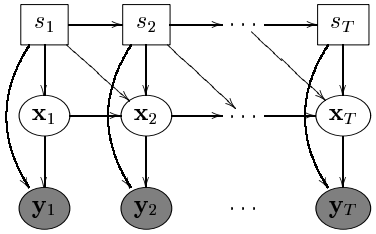



Fig. 2. A switching linear dynamical system. The dynamic switches $s_t$ determine both the dynamics in the continuous latent variables $\mathbf{x}_t$ and the link between $\mathbf{x}_t$ and the observations $\mathbf{y}_t$.

*ing linear dynamical system* (SLDS) that allows projection onto multiple plots. That is, the SLDS describes the data as if generated from a *dynamic mixture* of two-dimensional hyperplanes. Alternatively, the model can be seen as a dynamic extension of the mixtures of PPCA model, where now both the continuous and discrete latent variables are dynamic rather than static.

With every observation $\mathbf{y}_t$ we associate a continuous latent variable $\mathbf{x}_t$, tracking the *state* of the system, and a discrete latent variable $s_t$, representing the current *regime*. Within every regime the state follows a simple LDS. The regime itself is governed by a first-order Markov process.

The complete model that we consider here reads

$$\begin{aligned} p(s_{t+1} = j | s_t = i, \boldsymbol{\theta}) &= \Pi_{i \to j} \\ p(\mathbf{x}_{t+1} | \mathbf{x}_t, s_t = i, s_{t+1} = j, \boldsymbol{\theta}) & \\ &= \begin{cases} \mathcal{N}(\mathbf{x}_{t+1}; A_j \mathbf{x}_t, Q_j) : i = j \\ \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{0}, \Sigma_j) : i \neq j \end{cases} \\ p(\mathbf{y}_t | \mathbf{x}_t, s_t = i, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y}_t; C_i \mathbf{x}_t + \boldsymbol{\mu}_i, r_i^2 I) \; . \end{aligned} \tag{1}$$

The corresponding Bayesian network representation is visualized in Figure 2.

Any of the approaches described in Appendix A can be taken to define priors for $\mathbf{x}_1 | s_1 = i$ and $s_1$ itself. As can be seen from (1), $\mathbf{x}_t$, the state of the system, is 'reset' after a regime switch and a new state is drawn from a Gaussian prior. This reset effectively decouples the latent axes of the different regimes. We can take the mean of the Gaussian prior equal to $\mathbf{0}$ without loss of generality.

In principle every observation is projected onto every plot at its corresponding posterior

mean, e.g. for plot $i$: $E[\mathbf{x}_t|s_t = i, \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\mathrm{ML}}]$. The regime posterior $p(s_t = i|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\mathrm{ML}})$ can then be taken as the amount of 'ink' that is to be used [3]. Alternatively, one can choose to plot only those projections that have $p(s_t = i|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\mathrm{ML}})$ larger than a preset threshold. Recall that in the dynamic model the posterior does not factorize, so in contrast to [3] we have $p(s_t = i|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \neq p(s_t|\mathbf{y}_t, \boldsymbol{\theta})$.

The ML-estimate $\boldsymbol{\theta}_{\mathrm{ML}}$, which includes all parameters of the model ($\{C_i, \boldsymbol{\mu}_i, r_i^2, A_i, Q_i\}$ for all $i$, the transition matrix $\Pi$ and perhaps extra parameters for specifying the priors), can again be found using an EM-procedure. This, however, is somewhat more complicated then for the LDS presented above. A first difficulty is that straightforward ML estimation may cause problems: $\Pi_{i\to j}$ can become 0 if certain regime transitions do not occur, which in general is undesirable. Also if there are many different regimes, $r_i^2$'s may reduce to 0. These problems can be prevented by using conjugate priors and computing MAP estimates.

A second, more important problem is the selection of an appropriate number of regimes $M$ and an initial parameter setting for EM. These are well-known problems for Gaussian mixtures and also occur in the SLDS model. For data exploration Bishop and Tipping [3] propose an interactive approach, which we adopt here: the data is first projected onto a single plot, after which the user selects the number of regimes and their approximate center locations using mouse clicks. The clicked locations can be projected back onto the high-dimensional space. There, a rough clustering and plane fitting routine provides initial estimates for EM. As for the LDS, we initialize $A_j = 0$, i.e. the initial parameter settings are identical to the mixture of PPCAs model in [3].

The last and most difficult problem is that exact inference in a general SLDS is NP-hard [4], i.e. computing the required posteriors would take time and memory requirements exponential in the number of observations. Strictly speaking the reset after a regime switch makes the complexity of inference polynomial in the length of the sequence[2], but this is still unacceptable for a data exploration application. In Section IV we introduce

[2] Only when a regime started and when it ended is relevant. Differences in regime history before a reset do not result in a different mode in a smoothed posterior. The exact posteriors for the first and last time slice have only $MT$ different modes. Posteriors for slices in the middle of the sequence have $\mathcal{O}(MT^2)$ modes. Since the running time of the Kalman smoother is linear in $T$, treating all other quantities as a constant, exact inference could be performed in $\mathcal{O}(T^4)$ time.

an approximate inference (E-step) scheme that fits very well with the characteristics of the visualization problem. The M-step is straightforward and can be computed without further approximations given the approximate inference results (see Appendix C for the update equations). Although the complexity of the complete algorithm is slightly higher than that of a mixture of PPCAs it scales linearly with $d$, the dimensionality of the observations, and $T$, the length of the sequence. Therefore we expect it to be appropriate in all the situations that the static model can be used (for further details see Appendix B).

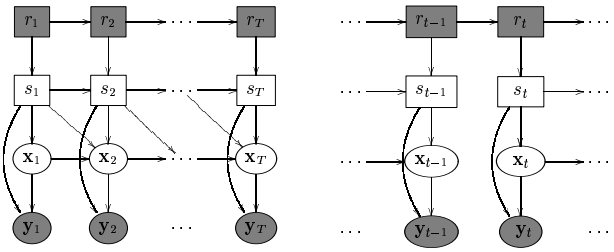
## III. DIFFERENT LEVELS OF DETAIL

The projection onto a single plot can be seen as the top level of a hierarchy of plots. An example is given in Figure 6. The plots produced by the SLDS form a second level. We can extend the SLDS model by adding a third and, recursively, subsequent levels. The top plot then gives a coarse grained overview of the data, allowing the user to interactively zoom in on non-trivial phenomena in the subtrees. Such an approach is taken by [3] for a mixture of PPCAs. In this section we show that with appropriate constraints, the same can be done for time-series data.

The model for the third level has to ensure that the first two levels are extended into a hierarchy, e.g. in Figure 6 the third level is not the result of an arbitrary SLDS with 9 possible regimes, but constructed such that regimes (plots) refine regimes from the second level that were already studied and identified by the user. A second requirement for the new model is that when we expand the hierarchy to have many levels both the computational complexity and the number of free parameters should stay within reasonable bounds. We will describe a model that meets the first requirement in Section III-A and refine it in Section III-B such that the second is also met.

### *A. Constrained inference*

The generative model that extends the second level is visualized in Figure 3(a). It is essentially also an SLDS and represents a collection of plots at the third level. Note that this model *only* generates the plots for the third level. Within this section we will use $s_{1:T}$ to refer to the switches for the plots in the third layer. To avoid confusion we use $\boldsymbol{\theta}^{\text{par}}$ and $r_{1:T}$ to refer to the parameters and switches from the model that produced the plots




(a) General case (b) When $r_{t-1} \neq r_t$.

Fig. 3. Graphical structure corresponding to the third and subsequent levels of the hierarchical switching linear dynamical system.

for the second layer (Section II-B, Figure 2), which are fixed at this stage. By clicking at locations in a parent plot the user determines the number of child plots and approximate initial estimates for EM, as sketched in Section II-B. We refer to the implied parent-child relationship with pa(.), i.e. $\mathrm{pa}(j) = m$ is true if and only if subplot $j$'s initial parameter settings for EM were computed based on the user's mouse-click in plot $m$. Similarly, we use $\mathrm{ch}\,(m)$ to denote all children of $m$.

To obtain a proper hierarchy all points in the parent should be visible in some child, but no new points should reappear. The strictest choice of requirements that enforces this is:

$$\sum_{s_{1:T} \in \mathrm{ch}(r_{1:T})} p(s_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{\mathrm{par}}) \ , \tag{2}$$

where the shorthand is defined as $s_{1:T} \in \mathrm{ch}\,(r_{1:T})$ iff for all $s_t$: $s_t \in \mathrm{ch}\,(r_t)$. It is perhaps important to emphasize that the approach of fixing $p(r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{\mathrm{par}})$ is motivated by the fact that we do not want to change the plots from the second level once these have been shown to the user. In a different setting, e.g. if inference was done on a globel model, jointly over both the second and the third level, the posteriors over $s_{1:T}$ and $r_{1:T}$ might be different. This is due to the fact that, if we define a parent-child relationship, regimes at the third level influence regimes at the second and vice versa.

We will first show how to satisfy (2) in the case that there is a known sequence $r_{1:T}$. In this case the constraint (2) simplifies to

$$\sum_{s_{1:T} \in \mathrm{ch}(r_{1:T})} p(s_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) = 1 \ , \tag{3}$$

To be able to enforce constraints on *sums* of $s_{1:T}$ we extend the state space with the switch states of the parent level (see Figure 3). The link connecting $r_t$ with $s_t$ is defined to satisfy

$$p(s_{t+1}|s_t = i, r_{t+1}, \boldsymbol{\theta}) = 0 \text{ , if } s_{t+1} \notin \text{ch}\,(r_{t+1}). \tag{4}$$

This ensures that, for every history $s_{1:T}$ with at least one $s_t \notin \text{ch}\,(r_t)$, the conditional posterior $p(s_{1:T}|r_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta})$ is 0. And, therefore, we get $\sum_{s_{1:T} \in \text{ch}(r_{1:T})} p(s_{1:T}|r_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = 1$, as required. So in the case that $r_{1:T}$ is known, we can simply treat the extra switches as regular observations. Since in practice we observe that the posterior probabilities are close to being crisp, simply rounding off the posteriors at the parent level, and clamping $r_{1:T}$ would give a reasonable approximation.

In the general case we cannot treat $r_{1:T}$ as observed. Instead, we keep its marginal fixed to $p(r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{\text{par}})$ (*soft-clamping*). We will use the shorthand $p^*(r_{1:T}) \equiv p(r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{\text{par}})$. Soft clamping is performed using Jeffrey's rule of conditioning. In the current setting the marginal on $r_{1:T}$ is fixed by:

$$p^*(r_{1:T}, s_{1:T}, \mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \equiv p(s_{1:T}, \mathbf{x}_{1:T}|r_{1:T}, \mathbf{y}_{1:T}) p^*(r_{1:T}) \text{ ,} \tag{5}$$

The fact that $r_{1:T}$ are soft-clamped cannot be denoted in the same way as the hard clamped $\mathbf{y}_{1:T}$ (e.g. distributions over $r_{1:T}$ are still defined as in (5)). We will use $p^*(\cdot)$ for probabilities that have the constraints on $r_{1:T}$ enforced.

Since (4) enforces the child parent relationship, for a given sequence $s_{1:T}$, only the sequence $r_{1:T} = \text{pa}(s_{1:T})$ has $p(s_{1:T}, r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \neq 0$. So we have:

$$\sum_{r_{1:T}} p(s_{1:T}, r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(s_{1:T}, r_{1:T} = \text{pa}(s_{1:T})|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \text{ .} \tag{6}$$

Hence the posterior over $s_{1:T}$ has the property

$$\begin{aligned} \sum_{s_{1:T} \in \text{ch}(r_{1:T})} p^*(s_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) &= \sum_{s_{1:T} \in \text{ch}(r_{1:T})} p^*(s_{1:T}, r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \quad \text{(by (6))} \\ &= \sum_{s_{1:T} \in \text{ch}(r_{1:T})} p(s_{1:T}|r_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) p^*(r_{1:T}) \quad \text{(by (5))} \\ &= p^*(r_{1:T}) \quad \text{(by (4))} \text{ .} \end{aligned}$$

So the third level model can be made to agree with the second level model in the sense of (2). However, this cannot be done in reasonable time, since the constraints are over

$(M^{\text{par}})^T$ possible paths $r_{1:T}$, where $M^{\text{par}}$ is the number of plots at the parent level. In Section IV-A we show how a reasonable approximation can be made.

### *B. Independent sub-models*

We will now address the second requirement of the model and show that a reasonable choice for the regime transition probabilities $p(s_{t+1}|s_t, r_{t+1}, \boldsymbol{\theta})$ allows a depth first expansion of the tree and prevents a blow up of free parameters. Recall that a switch between *any* two regimes renders the current state $\mathbf{x}_t$ independent of $\mathbf{x}_{t-1}$ (1), so updates for the "inter-slice" parameters $A$ and $Q$ will not depend on statistics from other subtrees. To get a similar property for the transition probabilities we restrict the probabilities of jumps between plots from different subtrees ("cousins"):

$$p(s_{t+1} = j|s_t = i, r_{t+1} = l, \boldsymbol{\theta}) \tag{7}$$
$$= \begin{cases} 0 & : \quad \text{pa}(j) \neq l & (7-a) \\ \Pi_{i\to j|l} & : \quad \text{pa}(i) = \text{pa}(j) = l & (7-b) \\ \pi_{j|l} & : \quad \text{pa}(i) \neq l = \text{pa}(j)\,. & (7-c) \end{cases}$$

Case (7-a) encodes the above introduced constraint that every child has only one parent. Case (7-b) states that if we stay within the same subtree, the probability of jumping between plots within one subtree ("brothers") is fully modelled. Case (7-c) states that if there is a jump on the parent level, the new child is drawn from a prior. Since we (soft) clamp the values of $r_{1:T}$ there is no direct need for a definition of transition probabilities $p(r_{t+1}|r_t, \boldsymbol{\theta})$. To define a self-contained model we define $p(r_{t+1}|r_t, \boldsymbol{\theta}) \equiv p(r_{t+1}|r_t, \boldsymbol{\theta}^{\text{par}})$. The arcs in Figure 6 show examples of the different transitions; the learned transition probabilities are plotted in Figure 5.

The specified model now has the property that $\{s_{t-1}, \mathbf{x}_{t-1}\}$ and $\{s_t, \mathbf{x}_t\}$ uncouple in the case of a switch between subtrees (Figure 3(b)). The conditional independencies in the

model allow the posterior to be written as:

$$
\begin{aligned}
p(s_{1:T}, \mathbf{x}_{1:T} | \mathbf{y}_{1:T}, r_{1:T}, \boldsymbol{\theta}) \;\; &\propto \;\; \prod_{\substack{t=2 \\ r_{t-1}=r_t}}^{T} p(\mathbf{y}_t, \mathbf{x}_t, s_t | \mathbf{x}_{t-1}, s_{t-1}, r_t, \boldsymbol{\theta}) \prod_{\substack{t=1 \\ r_{t-1}\neq r_t}}^{T} p(\mathbf{y}_t, \mathbf{x}_t, s_t | r_t, \boldsymbol{\theta}) \\
&= \prod_{m=1}^{M^{\text{par}}} \Bigg\{ \prod_{\substack{t=2 \\ r_{t-1}=r_t=m}}^{T} p(\mathbf{y}_t, \mathbf{x}_t, s_t^{(m)} | \mathbf{x}_{t-1}, s_{t-1}^{(m)}, r_t = m, \boldsymbol{\theta}^{(m)}) \\
&\qquad \prod_{\substack{t=1 \\ r_{t-1}\neq r_t=m}}^{T} p(\mathbf{y}_t, \mathbf{x}_t, s_t^{(m)} | r_t = m, \boldsymbol{\theta}^{(m)}) \Bigg\} \;, \qquad (8)
\end{aligned}
$$

where $\boldsymbol{\theta}^{(m)}$ are the disjoint parameter sets that together form $\boldsymbol{\theta}$:

$$
\boldsymbol{\theta}^{(m)} = \left\{ \pi_{j|m}, \Pi_{i \to j|m}, A_j, Q_j, \Sigma_j, C_j, \boldsymbol{\mu}_j, r_j^2 | \text{pa}(i) = \text{pa}(j) = m \right\} \;,
$$

and $s_t^{(m)}$ are variables ranging over ch $(m)$. The boundary $t = 1$ can be taken into account by setting $r_0 \neq m$ for all $m$.

If we use the conditional independencies in (8) to rewrite the expected log-likelihood $\hat{\mathcal{L}}$ we see that it factors into a sum of subtree terms:

$$
\begin{aligned}
\hat{\mathcal{L}}(\boldsymbol{\theta}) \;\; &= \;\; E_{p^*(s_{1:T}, \mathbf{x}_{1:T}, r_{1:T} | \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}})} \left[ \log p(r_{1:T}, s_{1:T}, \mathbf{x}_{1:T}, \mathbf{y}_{1:T} | \boldsymbol{\theta}) \right] \qquad (9) \\
&= \;\; const + \sum_{m=1}^{M^{\text{par}}} \sum_{t=1}^{T} \sum_{\substack{\{j | \text{pa}(j)=m\} \\ \{i | \text{pa}(i)=m\}}} E_{p_t^*\left(\begin{smallmatrix} \bar{m} m \\ \cdot j \end{smallmatrix}\right)} \left[ \log \pi_{j|m} \right] + E_{p_t^*\left(\begin{smallmatrix} m m \\ i j \end{smallmatrix}\right)} \left[ \log \Pi_{i \to j|m} \right] \\
&\quad + E_{p_t^*\left(\begin{smallmatrix} \cdot m \\ j j \end{smallmatrix}\right)} \left[ \log \mathcal{N}(\mathbf{x}_t; \mathbf{0}, \Sigma_j) \right] + E_{p_t^*\left(\begin{smallmatrix} m m \\ j j \end{smallmatrix}\right)} \left[ \log \mathcal{N}(\mathbf{x}_t; A_j \mathbf{x}_{t-1}, Q_j) \right] \\
&\quad + E_{p_t^*\left(\begin{smallmatrix} m \\ j \end{smallmatrix}\right)} \left[ \log \mathcal{N}(\mathbf{y}_t; C_j \mathbf{x}_t + \boldsymbol{\mu}_j, r_j^2 I) \right] \\
&\equiv \;\; const + \sum_{m=1}^{M^{\text{par}}} \hat{\mathcal{L}}^{(m)}(\boldsymbol{\theta}^{(m)}) \;.
\end{aligned}
$$

In the above *const* is a constant independent of $\boldsymbol{\theta}$ introduced by the $\log r_{1:T}$ term, and $p_\cdot^*(.)$ denote the appropriate one and two-slice posteriors:

$$
\begin{aligned}
p_t^*\left(\begin{smallmatrix} \bar{m} m \\ \cdot j \end{smallmatrix}\right) &\equiv p^*(r_{t-1} \neq m, r_t = m, s_t = j | \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) \qquad (10) \\
p_t^*\left(\begin{smallmatrix} m m \\ i j \end{smallmatrix}\right) &\equiv p^*(r_{t-1} = r_t = m, s_{t-1} = i, s_t = j, \mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) \qquad (11) \\
p_t^*\left(\begin{smallmatrix} \cdot m \\ j j \end{smallmatrix}\right) &\equiv p^*(r_t = m, s_{t-1} \neq j, s_t = j, \mathbf{x}_t | \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) \qquad (12) \\
p_t^*\left(\begin{smallmatrix} m \\ j \end{smallmatrix}\right) &\equiv p^*(r_t = m, s_t = j, \mathbf{x}_t | \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) \;, \qquad (13)
\end{aligned}
$$

where at the boundary $t = 1$ we define $p_1^*\left(\begin{smallmatrix} m & m \\ i & j \end{smallmatrix}\right) \equiv 0$ and $p_1^*\left(\begin{smallmatrix} \bar{m} & m \\ \cdot & j \end{smallmatrix}\right) \equiv p_1^*\left(\begin{smallmatrix} \cdot & m \\ \bar{j} & j \end{smallmatrix}\right) \equiv p_1^*\left(\begin{smallmatrix} m \\ j \end{smallmatrix}\right)$.

The marginal $p^*(r_{1:T})$ is fixed, so by (8) the required posterior $p^*(s_{1:T}, \mathbf{x}_{1:T}, r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}})$ in (9) factors into independent subtree terms. Therefore these terms, and hence the required statistics (10) to (13), can be computed independently. Furthermore, changes in $\boldsymbol{\theta}^{(m)}$ *only* affect the posterior terms corresponding to $m$, so $\mathcal{L}^{(n)}$ with $n \neq m$ remains unchanged.

The normalization constraints $\sum_j \pi_{j|m} = 1$ and $\sum_j \Pi_{i \to j|m} = 1$, by definition (7), can be rewritten as $\sum_{s^{(m)}} \pi_{j|m} = 1$ and $\sum_{s^{(m)}} \Pi_{i \to j|m} = 1$. Combining above observations we conclude that we can maximize $\hat{\mathcal{L}}$ w.r.t. $\boldsymbol{\theta}$ by independently maximizing every $\hat{\mathcal{L}}^{(m)}$ w.r.t. $\boldsymbol{\theta}^{(m)}$ under local normalization constraints, i.e. subtrees can be expanded independently depth-first.

In practice for many time slices $p^*(r_t = m)$ will be close to 0, so the amount of "ink" attributed to $\mathbf{y}_t$ in any subplot in $m$ will be so low that $\mathbf{y}_t$ is never projected in subtree $m$. We can therefore safely consider only those sub-sequences of $\mathbf{y}_{1:T}$ that have significant weight in parent $m$ of the second layer. This results in a considerable speed-up.

The posteriors $p^*(s_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{ML}})$ form the constraints for a subsequent layer. This way, the user can keep adding layers of subplots recursively, until all relevant clusters can be inspected and interpreted.

## IV. APPROXIMATE INFERENCE IN AN SLDS

With the above choices, many of the appealing features of the static versions of (hierarchical) PPCA apply to the dynamic extension proposed in this article. There is one important difference: exact inference in the E-step of the EM-algorithm is practically infeasible for switching linear dynamical systems. One way to see this is to consider posteriors over the state of the system, e.g. $p(\mathbf{x}_t|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}})$. In a general SLDS this posterior has $M^T$ modes:

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) = \sum_{s_{1:T}} p(\mathbf{x}_t|s_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) p(s_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}) \ ,$$

one mode for every possible regime history $s_{1:T}$. Although strictly speaking the reset dynamics in the current model reduces the complexity to become polynomial (see the footnote on page 7), this is in an interactive visualization application still unacceptable.

In this section we therefore describe a greedy approximate inference strategy for the SLDS. An extension to the model for the third and subsequent levels is presented in Section IV-A. The algorithm can be interpreted as approximate belief propagation and is a particular form of *expectation propagation* [5]. The filtering pass of this approximation has been proposed independently several times. The oldest reference we are aware of is in [6], in the engineering literature it is known as *generalized pseudo Bayes 2* (GPB2) [7]. The basic idea is to approximate posteriors such as $p(\mathbf{x}_t|s_t = i, \mathbf{y}_{1:T})$ (where for ease of notation we leave out the dependency on the model parameters $\boldsymbol{\theta}_{\text{old}}$) with a single Gaussian instead of keeping the exact mixture. In the context of visualization, where we are only interested in the mean, this is well justified. Our method works for a general SLDS, but benefits for this particular model automatically from the simplification introduced by the 'reset' after a regime switch.

The intuition behind the method is best understood by considering the filtering pass. The exact posterior $p(\mathbf{x}_1, s_1|\mathbf{y}_1)$ is *conditionally Gaussian* (CG) distributed; conditioned on $s_1$, $\mathbf{x}_1$ is Gaussian distributed. However the posterior

$$p(\mathbf{x}_2, s_2|\mathbf{y}_{1:2}) = \int_{d\mathbf{x}_1} \sum_{s_1} p(\mathbf{x}_1, \mathbf{x}_2, s_1, s_2|\mathbf{y}_{1:2})$$

is *not* conditionally Gaussian anymore, but of a more complex form; conditioned on $s_2 = i$, $\mathbf{x}_2$ is a mixture of two Gaussians (one for the case $s_1 \neq i$ and one for $s_1 = i$). Instead of using this exact form to recursively compute the posterior for the third time slice (and thus introducing a rapid growth of complexity in time) we first approximate it by $q(\mathbf{x}_2, s_2)$, the conditionally Gaussian distribution closest in Kullback-Leibler (KL) divergence to the original mixture, where

$$\mathrm{KL}\left(p||q\right) \equiv \sum_{s_2} \int_{d\mathbf{x}_2} p(\mathbf{x}_2, s_2|\mathbf{y}_{1:2}) \log \frac{p(\mathbf{x}_2, s_2|\mathbf{y}_{1:2})}{q(\mathbf{x}_2, s_2)} \ .$$

It can easily be shown that minimizing the above KL divergence boils down to *moment matching* or a *collapse* of the mixture (see e.g. [8]). We will use the notation

$$\text{Collapse}\left(\sum_{\text{s}} \text{p(s, r, x)}\right) \equiv \underset{\text{q}\in\text{CG}}{\text{argmin}}\, \text{KL}\left(\sum_{\text{s}} \text{p(s, r, x)}||\text{q(r, x)}\right) \ .$$

Define $p(s = i, r = j, \mathbf{x}) = p_{ij}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ij}, \Sigma_{ij})$, then $\text{Collapse}(\sum_{\text{s}} \text{p(s, r, x)}) = \text{q(r, x)}$ with $q(r = j, \mathbf{x}) = p_j\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \Sigma_j)$, $p_j = \sum_i p_{ij}$, $\boldsymbol{\mu}_j = \sum_i p_{i|j}\boldsymbol{\mu}_{ij}$, $\Sigma_j = \sum_i p_{i|j}(\Sigma_{ij} + (\boldsymbol{\mu}_{ij} -$

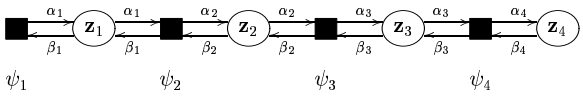



Fig. 4. Message propagation.

$\boldsymbol{\mu}_j)(\boldsymbol{\mu}_{ij} - \boldsymbol{\mu}_j)^T)$, and $p_{i|j} = p_{ij}/p_j$. A recursive filtering procedure based on such a collapse keeps $M$ Gaussians, one for each switch state, in every time slice.

The algorithm we propose here can be seen as the extension of the above scheme with an analogous smoothing pass. The presentation is in spirit of the sum-product algorithm [9]. The posterior distribution can be written as a product of *local potentials*, $\psi_t$:

$$\begin{aligned} p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}) &\propto \prod_{t=1}^{T} \psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \\ \psi_1(\mathbf{z}_0, \mathbf{z}_1) &\equiv p(\mathbf{y}_1|\mathbf{z}_1) \\ \psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t) &\equiv p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1}), \quad t > 1 \ . \end{aligned}$$

For ease of notation and interpretability we treat $\mathbf{z}_t = \{s_t, \mathbf{x}_t\}$ together as one CG distributed random variable. Slightly abusing notation, we will use the sum sign for the combined operation of summing out $s_t$ and integrating out $\mathbf{x}_t$. We are interested in one and two-slice marginals of the joint posterior. To avoid having to construct the entire posterior, these marginals are computed by local operations where *messages* are sent between nodes in a graph. We distinguish *variable nodes* that are associated with $\mathbf{z}_t$'s and *function nodes* that are associated with $\psi_t$'s. The message from $\psi_t$ forward to $\mathbf{z}_t$ is called $\alpha_t(\mathbf{z}_t)$ and the message from $\psi_t$ back to $\mathbf{z}_{t-1}$ is referred to as $\beta_{t-1}(\mathbf{z}_{t-1})$. In a chain, variable nodes simply pass on the messages they receive. The message passing scheme is depicted in Figure 4.

We denote the approximation of $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ by $q_t(\mathbf{z}_t)$. It is computed by multiplying all incoming messages from neighboring function nodes: $q_t(\mathbf{z}_t) \propto \alpha_t(\mathbf{z}_t)\beta_t(\mathbf{z}_t)$. Associated with every function node is an approximate two-slice belief that we denote by $p(\mathbf{z}_{t-1}, \mathbf{z}_t|\mathbf{y}_{1:T}) \approx \hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t)$. Now new messages from function node $\psi_t$ to variable node $\mathbf{z}_{t'}$, where $t'$ can be $t-1$ or $t$, are computed as follows:

1. Construct a two-slice belief by multiplying the potential corresponding to the local function node $\psi_t$ with *all* messages from neighboring variable nodes to $\psi_t$, yielding

$$\hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \propto \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t) \ .$$

2. Find the one-slice marginal $q_{t'}(\mathbf{z}_{t'})$ which is conditionally Gaussian but approximates $\hat{p}_t(\mathbf{z}_{t'})$ best in KL sense by:

$$q_t(\mathbf{z}'_t) \propto \text{Collapse}\left(\sum_{\mathbf{z}_{t''}} \hat{\mathrm{p}}_{\mathrm{t}}(\mathbf{z}_{t''}, \mathbf{z}_{\mathrm{t}})\right) ,$$

where $t''$ is $\{t-1, t\} \backslash t'$.

3. Infer the new message by division. All messages not sent from $\psi_t$ remain fixed, in particular $\beta_t$ and $\alpha_{t-1}$, so new messages are computed as:

$$\alpha_t(\mathbf{z}_t) = \frac{q_t(\mathbf{z}_t)}{\beta_t(\mathbf{z}_t)} , \quad \beta_{t-1}(\mathbf{z}_{t-1}) = \frac{q_{t-1}(\mathbf{z}_{t-1})}{\alpha_{t-1}(\mathbf{z}_{t-1})} .$$

If all messages are initialized with $\mathbf{1}$ the first forward pass of this algorithm performs the filtering scheme sketched above. Readers familiar with the HMM can easily verify that if we identify $\alpha_{t-1}(\mathbf{z}_{t-1}) \propto p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})$ and $\beta_t(\mathbf{z}_t) \propto p(\mathbf{y}_{t+1:T}|\mathbf{z}_t)$ this algorithm is equivalent to the familiar forward-backward algorithm if applied to the fully discrete HMM (where a collapse is not necessary and the KL minimization results in exact marginalization). Similarly the algorithm would be equivalent to the two filter form of the forward backward algorithm if used for the Kalman filter model.

An important difference is that in the HMM and the Kalman filter case the forward and backward passes can be done independently, since on the forward pass multiplying by $\beta_t$ in step 1 and dividing again by $\beta_t$ in step 3 is superfluous since "multiplication + marginalization + division = marginalization". However "multiplication + collapse + division $\neq$ collapse", hence for the SLDS the forward and backward passes cannot be done independently, changes in $\beta$'s on the backward pass would result in different $\alpha$'s in a new forward pass and vice versa for changes in $\alpha$'s. We should therefore iterate forward and backward passes. Pseudo code for the approximate inference procedure is given in Algorithm 1. The "proportional to" assignment ($\propto$) is defined as the assignment of the r.h.s. to the l.h.s. *after a normalization.* At the boundaries we keep messages $\alpha_0 = 1$ and $\beta_T = 1$ throughout the procedure, or alternatively we treat the $t = 1$ in line 9 and the $t = T$ in line 14 as special cases.

Fixed points of Algorithm 1 correspond to fixed points of a "Bethe free energy" [10]:

$$\mathcal{F}_{\mathrm{EP}}(\hat{p}, q) = \sum_t \sum_{\mathbf{z}_{t-1,t}} \hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \log \frac{\hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t)}{\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)} - \sum_t \sum_{\mathbf{z}_t} q_t(\mathbf{z}_t) \log q_t(\mathbf{z}_t) \ , \qquad (14)$$

subject to the constraints that all $\hat{p}_t$'s and $q_t$'s sum to 1, and "weak" consistency constraints:

$$\mathrm{Collapse}\left(\sum_{\mathbf{z}_{t-1}} \hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\right) = q_t(\mathbf{z}_t) = \mathrm{Collapse}\left(\sum_{\mathbf{z}_{t+1}} \hat{p}_{t+1}(\mathbf{z}_t, \mathbf{z}_{t+1})\right) \ .$$

This relationship is analogous to the one between the Bethe free energy and loopy belief propagation [11]. The only difference is that the strong consistency constraints are replaced by the weak ones: i.e. here overlapping beliefs only have to agree on their *expectations*. The proof of this claim is similar to the one in [11] and follows by constructing the Lagrangian and setting its derivatives to 0. In the resulting stationary conditions the Lagrange multipliers added for the weak consistency constraints have a one-to-one correspondence with messages $\alpha_t$ and $\beta_t$: the multipliers form the canonical parameters of the messages. Given this relationship the mapping between fixed points of the message passing scheme and stationary of points of $\mathcal{F}_{\mathrm{EP}}$ follows easily.

So Algorithm 1 can be seen as a procedure that greedily tries to find one and two-slice marginals that approximate the exact beliefs as good as possible and are pairwise consistent *after a collapse*.

Full details of the approximation are beyond the scope of this text. We refer the interested reader to [12]. This article presents an algorithm guaranteed to converge to a minimum of $\mathcal{F}_{\mathrm{EP}}$ and discusses stability properties of expectation propagation. It also provides extensive simulations to show empirically the validity of the method.

### *A. APPROXIMATE INFERENCE WITH CONSTRAINTS*

The model for the third and subsequent layers has additional variables $r_{1:T}$ to enforce the hierarchy interpretation of the visualization method. In the inference step we keep the marginal over $r_{1:T}$ fixed to $p^*(r_{1:T}) = p(r_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta}^{\mathrm{par}})$. We are interested in one and two-slice marginals of $p^*(r_{1:T}, s_{1:T}, \mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \boldsymbol{\theta})$ as defined in (5). Just as with inference in the SLDS without constraints this problem cannot be tackled in a reasonable time. In

**Algorithm 1** Approximate inference in an SLDS

$\alpha_0 \Leftarrow 1$

**for** $t = 1$ to $T$ **do**

$\alpha_t(\mathbf{z}_t) \Leftarrow 1$

$\beta_t(\mathbf{z}_t) \Leftarrow 1$

$\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \Leftarrow \begin{cases} p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t) & : \quad t = 1 \\ p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1}) & : \quad t > 1 \end{cases}$

**end for**

**while** ¬converged **do**

**for** $t = 1$ to $T$ **do** {Filtering}

$\hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \propto \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)$

$q_t(\mathbf{z}_t) \Leftarrow \text{Collapse}\left(\sum_{\mathbf{z}_{t-1}} \hat{\mathrm{p}}_\mathrm{t}(\mathbf{z}_{\mathrm{t}-1}, \mathbf{z}_\mathrm{t})\right)$

$\alpha_t(\mathbf{z}_t) \Leftarrow \frac{q_t(\mathbf{z}_t)}{\beta_t(\mathbf{z}_t)}$

**end for**

**for** $t = T$ to 2 **do** {Smoothing}

$\hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) \propto \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)$

$q_{t-1}(\mathbf{z}_{t-1}) \Leftarrow \text{Collapse}\left(\sum_{\mathbf{z}_\mathrm{t}} \hat{\mathrm{p}}_\mathrm{t}(\mathbf{z}_{\mathrm{t}-1}, \mathbf{z}_\mathrm{t})\right)$

$\beta_{t-1}(\mathbf{z}_{t-1}) \Leftarrow \frac{q_{t-1}(\mathbf{z}_{t-1})}{\alpha_{t-1}(\mathbf{z}_{t-1})}$

**end for**

**end while**

this section we introduce the extension of Algorithm 1 that allows marginal constraints on $r_{1:T}$. We start out with the situation that the hierarchy is extended breadth-first, i.e. that the entire third layer is fitted jointly.

In the parent layer we have approximated the posterior by (weakly) consistent overlapping two-slice marginals. We use these results as constraints for the third layer, i.e. rather then requiring consistency as in (2), we restrict ourselves to consistency of overlapping two-slice marginals only:

$$p(r_{t-1,t}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) = p^*(r_{t-1}, r_t) \ ,$$

for $t = 2 : T$. This results in a free energy identical to (14) but now with definition

$\mathbf{z}_t \equiv \{r_t, s_t, \mathbf{x}_t\}$ and the constraints that all $\hat{p}_t$'s and $q_t$'s sum to one replaced by

$$\sum_{s_{t-1,t},\mathbf{x}_{t-1,t}} \hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) = p^*(r_{t-1}, r_t)$$

(proper normalization is automatically enforced since $\sum_{r_{t-1,t}} p^*(r_{t-1}, r_t) = 1$, also the weak consistency constraints ensure that the one-slice beliefs sum to $p^*(r_t)$). In the way new messages are computed only the first step needs to be changed

$1'$. Construct a two-slice belief that has the correct marginal over $r_{t-1,t}$ as follows:

$$\hat{p}_t(\mathbf{z}_{t-1}, \mathbf{z}_t) = \frac{\alpha_{t-1}(\mathbf{z}_{t-1})\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)}{\sum_{s_{t-1,t},\mathbf{x}_{t-1,t}} \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)} p^*(r_{t-1}, r_t) \ .$$

Fixed points of this new message passing scheme correspond to stationary points of $\mathcal{F}_{\mathrm{EP}}$ with the changed normalization constraints. The proof is analogous to the proof for the standard case presented in Section IV (see also [13]). The intuition behind the free energy is similar: the adapted message passing scheme tries to find one and two-slice marginals that approximate the exact beliefs as good as possible, are pairwise consistent after a collapse, *and* are consistent with the soft-assignments to parent plots from the previous layer.

The alternative message passing scheme based on $1'$ infers the required marginals for the model of the *entire* third layer. In Section III it was argued that inference can be done independently for every subtree $m$. We will now extend Algorithm 1 so that locally the marginals for subtree $m$ can be computed. One possible extension is to define random variables $s_t^{(m+)}$ that range over $j$ with $\mathrm{pa}(j) = m$ and an extra state $\bar{m}$, that encodes being in a subplot that is not a child of $m$. This would imply an inefficiency, since in the message passing scheme two-slice potentials are constructed with modes $\hat{p}_t(\bar{m}, \bar{m})$ that would never be used. Algorithm 2 presents the adaptation of Algorithm 1 where only the relevant parts of such a two-slice posterior are constructed. In Algorithm 2, $\mathbf{z}_t \equiv \{s_t^{(m)}, \mathbf{x}_t\}$, $w_t^{(mm)} \equiv p^*(r_{t-1} = r_t = m)$, and $w_t^{(\bar{m}m)} \equiv p^*(r_{t-1} \neq m, r_t = m)$. The statistics (10) to (13),

required in the M-step, can be retrieved from $\hat{p}_t$ and $q_t$ as follows:

$$\begin{aligned} p_t^*\left(\begin{smallmatrix}\bar{m}m\\ \cdot j\end{smallmatrix}\right) &= w_t^{(\bar{m}m)}\hat{p}_t^{(\bar{m}m)}(s_t = j) \\ p_t^*\left(\begin{smallmatrix}mm\\ ij\end{smallmatrix}\right) &= w_t^{(mm)}\hat{p}_t^{(mm)}(s_{t-1} = i, s_t = j, \mathbf{x}_{t-1}, \mathbf{x}_t) \\ p_t^*\left(\begin{smallmatrix}\cdot m\\ \bar{j}j\end{smallmatrix}\right) &= w_{t(\bar{j}j)}\hat{p}_t^{(\bar{m}m)}(s_t = j, \mathbf{x}_t) \\ p_t^*\left(\begin{smallmatrix}m\\ j\end{smallmatrix}\right) &= q_t(s_t = j, \mathbf{x}_t)\ , \end{aligned}$$

with $w_{t(\bar{j}j)} = q_t(s_t = j) - w_t^{(mm)}\hat{p}_t^{(mm)}(s_{t-1} = s_t = j)$.

**Algorithm 2** Approximate inference in subtree $m$

$\alpha_0 \Leftarrow 1$

**for** $t = 1$ to $T$ **do**

$\alpha_t(\mathbf{z}_t) \Leftarrow 1$

$\beta_t(\mathbf{z}_t) \Leftarrow 1$

$\psi_t^{(mm)}(\mathbf{z}_{t-1}, \mathbf{z}_t) \Leftarrow \begin{cases} p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t, r_t = m) & : \quad t = 1 \\ p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t, r_t = m|\mathbf{z}_{t-1}, r_{t-1} = m) & : \quad t > 1 \end{cases}$

$\psi_t^{(\bar{m}m)}(\mathbf{z}_t) \Leftarrow p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|r_{t-1} \neq m, r_t = m)$

**end for**

**while** ¬converged **do**

**for** $t = 1$ to $T$ **do** {Filtering}

$\hat{p}_t^{(mm)}(\mathbf{z}_{t-1}, \mathbf{z}_t) \propto \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t^{(mm)}(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)$

$\hat{p}_t^{(\bar{m}m)}(\mathbf{z}_t) \propto \psi_t^{(\bar{m}m)}(\mathbf{z}_t)\beta_t(\mathbf{z}_t)$

$q_t(\mathbf{z}_t) \Leftarrow \text{Collapse}\left(\mathrm{w}_\mathrm{t}^{(\mathrm{mm})} \sum_{\mathbf{z}_{\mathrm{t}-1}} \hat{\mathrm{p}}_\mathrm{t}^{(\mathrm{mm})}(\mathbf{z}_{\mathrm{t}-1}, \mathbf{z}_\mathrm{t}) + \mathrm{w}_\mathrm{t}^{(\bar{\mathrm{m}}\mathrm{m})}\hat{\mathrm{p}}_\mathrm{t}^{(\bar{\mathrm{m}}\mathrm{m})}(\mathbf{z}_\mathrm{t})\right)$

$\alpha_t(\mathbf{z}_t) \Leftarrow \frac{q_t(\mathbf{z}_t)}{\beta_t(\mathbf{z}_t)}$

**end for**

**for** $t = T$ to 2 **do** {Smoothing}

$\hat{p}_t^{(mm)}(\mathbf{z}_{t-1}, \mathbf{z}_t) \propto \alpha_{t-1}(\mathbf{z}_{t-1})\psi_t^{(mm)}(\mathbf{z}_{t-1}, \mathbf{z}_t)\beta_t(\mathbf{z}_t)$

$\hat{p}_{t-1}^{(m\bar{m})}(\mathbf{z}_{t-1}) \propto \alpha_{t-1}(\mathbf{z}_{t-1})$

$q_{t-1}(\mathbf{z}_{t-1}) \Leftarrow \text{Collapse}\left(\mathrm{w}_\mathrm{t}^{(\mathrm{mm})} \sum_{\mathbf{z}_\mathrm{t}} \hat{\mathrm{p}}_\mathrm{t}^{(\mathrm{mm})}(\mathbf{z}_{\mathrm{t}-1}, \mathbf{z}_\mathrm{t}) + \mathrm{w}_\mathrm{t}^{(\mathrm{m}\bar{\mathrm{m}})}\hat{\mathrm{p}}_{\mathrm{t}-1}^{(\mathrm{m}\bar{\mathrm{m}})}(\mathbf{z}_{\mathrm{t}-1})\right)$

$\beta_{t-1}(\mathbf{z}_{t-1}) \Leftarrow \frac{q_{t-1}(\mathbf{z}_{t-1})}{\alpha_{t-1}(\mathbf{z}_{t-1})}$

**end for**

**end while**

## V. PAPER MILL DATA

We present an example from a board production process. Board is made as a continuous web at production speeds ranging from 600 to 800 meters per minute. Multiple paper types are produced on the same line. The changes of type are performed without stopping production. The process tends to be complex, differing from machine to machine and, in general, is only partially understood. In this domain a visualization method may be useful both as a tool for novelty detection and as an on-line monitoring tool.

In Figure 6 ten hours of production data is presented. The data consists of 13 crucial sensor readings such as machine speed, steam pressures, flows of additives etc. measured once a minute. The gray scales in the figure indicate different types of paper (added after learning, i.e., this information is not used in any of the algorithms). The learned transition probabilities $p(s_{t+1}|s_t, \boldsymbol{\theta}_{\mathrm{ML}})$ at the third level are plotted in Figure 5. Figure 7 presents the same data, but is produced using PhiVis [3], the static counterpart of the SLDS hierarchy.

We see that macro scale clusters such as the paper type are well separated by both methods. The benefit of the dynamic model is more apparent in the lower levels where drifts (stretches with larger distances between projected points) and stable situations (closely spaced projections) in the state of the process can be observed. The bars below the subplots in Figure 6 show that there is a clear tendency to zoom in on data that is clustered in time (see for example the subdivision of the first parent plot at the second level into the children subplots at the third level). To enhance interpretability of the found clusters we experience that a simple interactive inspection tool helps greatly. In a two way system the user can select regions in individual sensor trends and see the corresponding projected points, the other way around the user can select projected points (usually a still unidentified cluster) by clicking a polygon around them and see the corresponding time points in the sensor trends.

## VI. RELATED WORK

Many different visualization algorithms have been proposed in the literature based on different assumptions about the data and hence are appropriate in different settings. The intended use of the hierarchy of linear dynamical systems as developed in this paper is

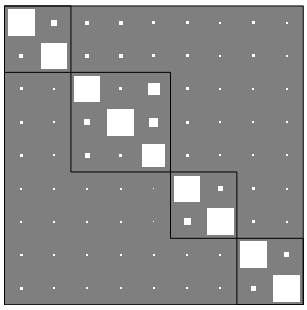


Fig. 5. Hinton diagram showing the learned transition probabilities $p(s_{t+1} = j|s_t = i, \boldsymbol{\theta}_{\mathrm{ML}})$ between the subplots at the third level of the hierarchy in Figure 6. In the Hinton diagram the surfaces of the squares are proportional to the corresponding probabilities.

an exploratory analysis of complex high-dimensional time series data, complex in the sense that multiple phenomenon occur on different levels of detail. Standard visualization procedures that explicitly assume the data to be independent and identically distributed could of course provide useful insights, but as is apparent from Figure 1, such studies can benefit from an additional analysis that incorporates the time structure.

Other methods developed for the visualization of high-dimensional time series data are extensions of the Self-Organizing Map (SOM) [14] and the Generative Topographic Mapping (GTM) through time model [15]. In the former an animated trajectory displays the series in a standard SOM. In the latter an extended GTM model is fitted. The GTM is essentially a probabilistic version of the SOM, in the variant for time-series the model is augmented with coarse Markov dynamics between latent (discrete) neurons.

Both methods fit a single non-linear 2D manifold in sensor space, the continuous latent variables are considered stationary, restricting the dynamics to transition probabilities between the nodes that they belong to. Furthermore, the structure of these models has to be determined in advance and does not give the user flexibility to interactively zoom in on the data. For simple processes, i.e. processes for which a single projection gives a clear description, the GTM through time could form an alternative monitoring tool. Another extension of the hierarchy of PPCAs, in an orthogonal direction, is presented in [16]. There, a hierarchy of (static) GTMs is described. This can be useful for datasets in which the non-linearity is stronger than can be captured by multiple locally linear models.

Hybrid dynamic Bayesian networks are widely studied and known by many names and

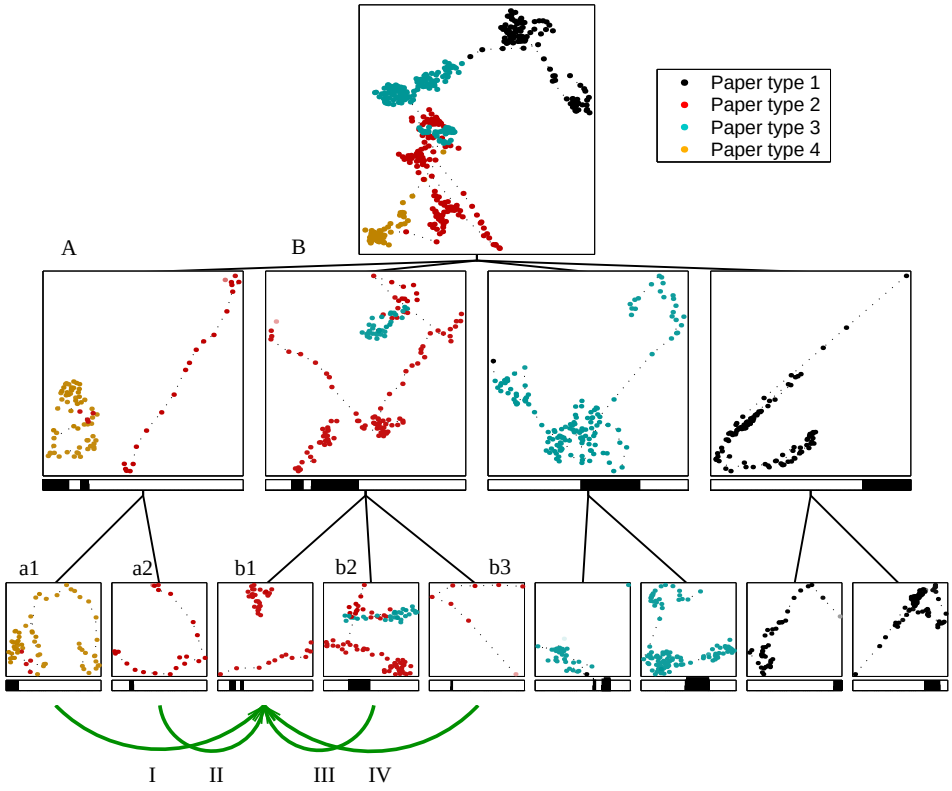



Fig. 6. Ten hours of production data from a paper mill projected using a hierarchy of switching linear dynamical systems. Dotted lines connect points in a trajectory. The color labels encode different paper types. Bars below the subplots visualize the probabilities $p(s_t|\mathbf{y}_{1:T})$ as a function of time. Transitions $I$; $p(b1|a1)$, and $II$; $p(b1|a2)$, are examples of jumps between cousin plots, by Eq. 7–$c$ these transitions are constrained to obey $p(b1|a1) = p(b1|B)p(B|A) = p(b1|a2)$. Transitions $III$; $p(b1|b2)$, and $IV$; $p(b1|b3)$ are jumps between brother plots. The corresponding transition probabilities are fully modeled (Eq. 7–$b$), hence in general $p(b1|b2) \neq p(b1|b3)$. Note that the labels themselves are not used by the algorithm and are added for clarity later. The number of child plots and their approximate starting locations are determined interactively by mouse-clicks as sketched in Section II-B.

variants. The intractability of exact inference is shared among all models that involve dynamics in a continuous latent space and discrete latent variables that either influence the corresponding transition, the observation model or both. In the formalism of Section IV where we define a two-slice potential that encapsulates the particular transition and observation models all variants of such hybrid models can be treated in a similar way.

The extension of the Kalman filter model to models with switching has been independently proposed in at least the econometrics (e.g. [6], [17]) and the tracking literature

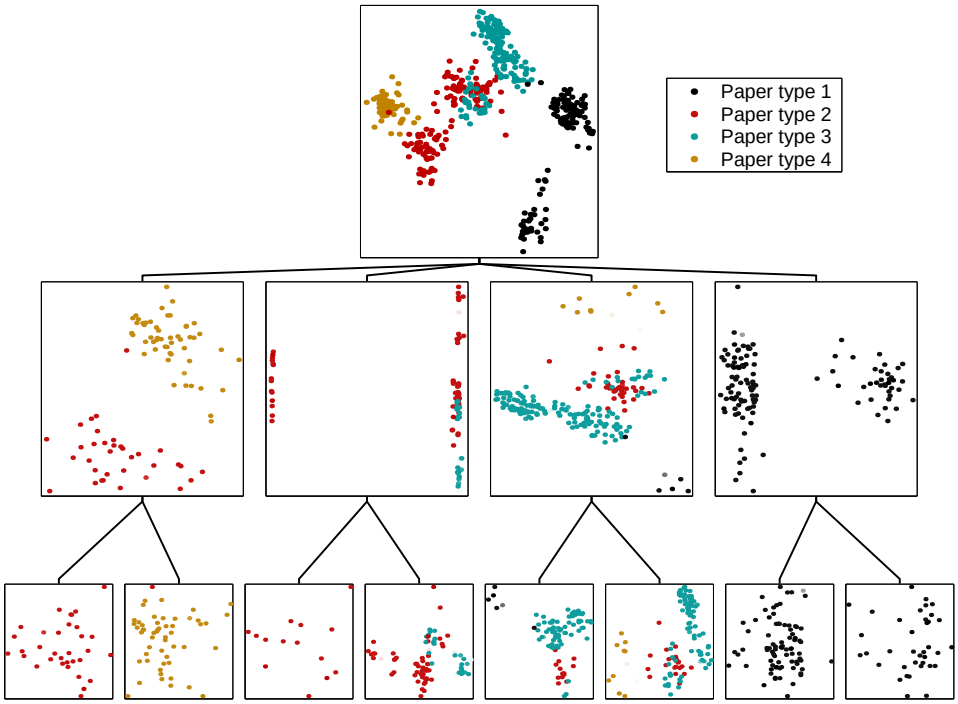



Fig. 7. The dataset from Figure 6 visualized using PhiVis; the static counterpart of the hierarchy of SLDS models. In the top plot mouse-clicks were at roughly the same locations as for Figure 6. Although most of the clusters that are visible in the SLDS projection are visible here, the dynamic structure is lost.

(e.g. [7] and references therein).

The first filtering pass of our algorithm is known as GPB2 in the tracking literature [7] and is also presented in e.g. [6]. In [17] a similar approximate filter and corresponding smoother is developed for a model where the discrete switches only govern the observation model. Contrary to what is suggested filtering under this restriction is not simpler, nor exact. The smoothing pass is also approximate: the smoothed posteriors over discrete variables are approximated by setting them to their filtered variants. The smoother proposed in [18] also keeps statistics fixed when smoothing. To our knowledge the EP algorithm described here and in [12] is the first algorithm to define a smoothing pass, which is symmetric to the GPB2 filter, and which does not introduce additional approximations.

Other variants of approximate inference include MCMC methods (e.g. [19]) and particle filter algorithms (e.g. [20]). Although these methods are useful in other applications, for the current visualization purposes they are not suited: the MCMC method is com-

putationally too expensive and the particle filter algorithm does not infer the smoothed statistics necessary for EM.

## VII. CONCLUSION

The visualization tool that we proposed in this article allows time series to be plotted in such a way that an approximation of the underlying structure within one time slice *and* between time slices can be presented. A hierarchy makes it possible to represent the process at different levels of detail. The applicability of our algorithm stems from the following crucial ingredients.

- Appropriate constraints in the SLDS corresponding to the third and subsequent levels. These constraints do not only enforce the "conservation of probability" going from a parent to its children, but also facilitate efficient inference and learning.
- An approximate inference algorithm for the E-step that scales linearly with the number of switches and the length of the sequence. The approximation strategy presented here, which is a specific case of expectation propagation, is particularly suited for the visualization of high-dimensional dynamic data with markedly different regimes, as encountered in our paper mill application.

Combination of the two makes the overall computational complexity essentially of the same order as its static counterpart, a hierarchical mixture of PPCAs (see Appendix B).

The visualization tool as presented here has great potential for exploration and on-line monitoring of high-dimensional dynamic data. Several natural and interesting extensions are possible. The current hierarchy represents a data set using multiple linear projections. Alternatively, one can, with a slight change in the model, display the data onto a single non-linear manifold. For example, if the system is not reset after a switch the orientation of the different locally linear manifolds are coupled. This might form an interesting alternative for the extra penalty term required in [21] to coordinate the local axes of factor analysis models. In principle, it should be possible to give a full (approximate) Bayesian treatment of all parameters involved, i.e., including the model parameters $\boldsymbol{\theta}$. Theoretically, such a model fits well in the EP-framework. However, the accuracy of such a method and a comparison with the simpler variational (mean-field) Bayes [22] or an MCMC based approach [23] has to be determined in practice. An extension to a model which models

dependencies between $s_{t+1}$ and $\mathbf{x}_t$ requires an adaptation of the approximate inference algorithm from Section IV due to the extra non-linearities [24], but may allow for better predictive capabilities.

## Acknowledgments

We acknowledge support from the Dutch Competence Centre Paper and Board and thank Ali Taylan Cemgil and Alexander Ypma for many helpful discussions and valuable comments. Parts of the drawing routines that produced Fig. 6 were based on PhiVis, the Matlab implementation accompanying [3]. PhiVis is available from `http://www.ncrg.aston.ac.uk/PhiVis/` under the GNU General Public License.

## Appendix

## I. PRACTICAL ISSUES

In this appendix we discuss several practical aspects that deserve special attention when implementing an LDS for visualization.

In a straightforward description of the LDS extra parameters are reserved for fitting the prior $p(\mathbf{x}_1)$ using maximum likelihood. Especially in the limit $A = 0$, but also for nonzero $A$, this can cause serious over-fitting, yielding rather unappealing projections of the first data point(s). There are several ways to resolve this (see e.g. [25]). One option is to take a fixed and broad (Gaussian) prior. This however treats the first data point different from the rest and does not reduce to PPCA with $A = 0$. Another, and perhaps more elegant solution is to take the stationary distribution of the latent-variable dynamics as a prior. This changes the M-step, since the transition matrix $A$ now also enters the log-likelihood through the prior $p(\mathbf{x}_1)$. Exact M-step updates can be obtained for constrained transitions $A = \alpha I$, relatively straightforward approximate ones for general $A$. Similarly, we can choose the prior $\pi$ for $s_1$ in the SLDS to be the leading eigenvector of the transition matrix $\Pi$, which again encodes the stationary distribution.

A second point of consideration is the fact that the model cannot be identified completely, i.e., the scale and orientation of the latent axes cannot be uniquely determined from the data. After convergence of the EM-algorithm we can turn the stationary distribution into a standard normal distribution and transform $C$ to have orthogonal columns.

The projection then has orthogonal axes with identical scales. Note that there are many possible orthogonal projections left, which explains the remaining rotational degree of freedom in Figure 1, but these are equivalent in terms of interpretability.

## II. COMPLEXITY

A standard implementation of the EM-algorithm for the LDS has complexity $\mathcal{O}(Tp^3)$ where $T$ is the number of observations and $p$ is the dimension of the observations. Here we treat the dimension of the latent space, which is two for visualization purposes, as a constant. The $d^3$ complexity is due to the inversion and determinant calculation of $p \times p$ matrices that all have the form $CVC^T + R$, with $C$ a $p \times q$ matrix, $V$ a positive definite $q \times q$ matrix and $R$ a positive constant times the identity matrix. Using

$$(CVC^T + R)^{-1} = R^{-1} - A^{-1}C(V^{-1} + C^T R^{-1} C)^{-1} C^T R^{-1} \,,$$

and

$$|CVC^T + R| = |R||I_q + C^T R^{-1} CV| \,,$$

the inversion and determinant calculation can be done in $\mathcal{O}(p)$ time. This makes it possible to compute both the posteriors and the likelihood of the model in $\mathcal{O}(Tp)$ time which is identical to the PPCA model. Similarly, treating $q$ as a constant the memory requirements are $\mathcal{O}(T)$ which is also identical to PPCA.

Approximate inference in a general SLDS using the algorithm in Section IV has complexity $M^2$ times that of the regular LDS algorithm, with $M$ the number of different values (regimes) $s_t$ can take. The reset after a regime switch however implies that the number of operations dealing with the high dimensional observations $d$ increases only by $2M$ (for all $i$, the approximations $\hat{p}_t(\mathbf{x}_{t-1}, \mathbf{x}_t, s_{t-1} = i, s_t = j)$ with $j \neq i$ are identical). To make use of this the construction of $\hat{p}_t$ in Algorithms 1 and 2 should be carefully implemented. The Markov process on the discrete latent states imply a summation over an $M^2$ transition matrix, but by our constraints in Section III, $M$ will stay within reasonable ranges. The final complexity of the SLDS and the model of the third layer is thus $\mathcal{O}(MTp + M^2 T)$, which we can expect to be reasonable for all situations in which the hierarchical PPCA model can be applied.

## III. M-step updates

In the M-step, $\hat{\mathcal{L}}(\boldsymbol{\theta})$ is maximized w.r.t. all parameters in $\boldsymbol{\theta}$. The updates can be found by adding Lagrange multipliers for the normalization constraints and setting derivates to 0. We give here the update equations for subtree $m$ in the model described in Section III-B. The LDS and SLDS updates follow as special cases.

The sufficient statistics (10) to (13) are calculated in the E-step. We use $\langle\cdot\rangle_{\cdot}$ to denote *weighted* expectations, e.g.

$$\begin{aligned}
&\left\langle f(\mathbf{x}_{t-1}, \mathbf{x}_t)\right\rangle_{p_t^*\binom{mm}{jj}} \\
&\quad = \quad p^*(s_{t-1} = s_t = j|\mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}^{(m)}) \int d\mathbf{x}_{t-1,t} f(\mathbf{x}_{t-1}, \mathbf{x}_t) p^*(\mathbf{x}_{t-1,t}|s_{t-1} = s_t = j, \mathbf{y}_{1:T}, \boldsymbol{\theta}_{\text{old}}^{(m)}) \ .
\end{aligned}$$

In this notation $\langle 1\rangle_{p_t^*\binom{mm}{jj}}$ simply gives the weighting factor. In the statistics above, and hence in the update equations below, we recognize forms similar to a regular LDS but now with a weighting term that would not be present in the non-switching case. The updates for $\pi_{j|m}$ and $\Pi_{i\to j|m}$ are weighted versions of the standard HMM updates.

$$\begin{aligned}
A_j^{\text{new}} &= \left(\sum_{t=2}^{T} \left\langle \mathbf{x}_t \mathbf{x}_{t-1}^T \right\rangle_{p_t^*\binom{mm}{jj}}\right) \left(\sum_{t=2}^{T} \left\langle \mathbf{x}_{t-1} \mathbf{x}_{t-1}^T \right\rangle_{p_t^*\binom{mm}{jj}}\right)^{-1} \\
Q_j^{\text{new}} &= \frac{\left(\sum_{t=2}^{T} \left\langle \mathbf{x}_t \mathbf{x}_t^T \right\rangle_{p_t^*\binom{mm}{jj}} - A_j^{\text{new}} \sum_{t=2}^{T} \left\langle \mathbf{x}_t \mathbf{x}_{t-1}^T \right\rangle_{p_t^*\binom{mm}{jj}}^T\right)}{\sum_{t=2}^{T} \langle 1\rangle_{p_t^*\binom{mm}{jj}}} \\
\Pi_{i\to j|m}^{\text{new}} &\propto \sum_{t=2}^{T} \langle 1\rangle_{p_t^*\binom{mm}{ij}} \\
\Sigma_j^{\text{new}} &= \frac{\sum_{t=1}^{T} \left\langle \mathbf{x}_t \mathbf{x}_t^T \right\rangle_{p_t^*\binom{\cdot m}{jj}}}{\sum_{t=1}^{T} \langle 1\rangle_{p_t^*\binom{\cdot m}{jj}}} \\
\pi_{j|m}^{\text{new}} &\propto \sum_{t=1}^{T} \langle 1\rangle_{p_t^*\binom{\bar{m}m}{\cdot j}}
\end{aligned}$$

We compute the new output matrix $C_j$ and the new mean $\boldsymbol{\mu}_j$ jointly by adding $\boldsymbol{\mu}_j$ as an extra column to $C_j$ and adding an entry to the continuous state that is always 1. We

define

$$
\begin{aligned}
P_{t,j} &\equiv \begin{bmatrix} \langle \mathbf{x}_t \mathbf{x}_t^T \rangle & \langle \mathbf{x}_t \rangle \\ \langle \mathbf{x}_t \rangle^T & \langle 1 \rangle \end{bmatrix} \\
\mathbf{m}_{t,j} &\equiv \begin{bmatrix} \langle \mathbf{x}_t \rangle \\ \langle 1 \rangle \end{bmatrix} \\
\tilde{C}_j^{\text{new}} &\equiv \begin{bmatrix} C_j^{\text{new}} & \boldsymbol{\mu}_j^{\text{new}} \end{bmatrix} ,
\end{aligned}
$$

with the weighted expectations $\langle \cdot \rangle$ over $p_t^* \binom{m}{j}$, to arrive at

$$
\begin{aligned}
\tilde{C}_j^{\text{new}} &= \left( \sum_{t=1}^{T} \mathbf{y}_t \mathbf{m}_{t,j}^T \right) \left( \sum_{t=1}^{T} P_{t,j} \right)^{-1} \\
r_j^{2\,\text{new}} &= \frac{\left( \sum_{t=1}^{T} \mathbf{y}_t^T \mathbf{y}_t \langle 1 \rangle_{p_t^* \binom{m}{j}} - \text{tr} \left[ \left( \tilde{C}_j^{\text{new}} \right)^T \left( \sum_{t=1}^{T} \mathbf{y}_t \mathbf{m}_{t,j}^T \right) \right] \right)}{p \sum_{t=1}^{T} \langle 1 \rangle_{p_t^* \binom{m}{j}}} ,
\end{aligned}
$$

where $p$ is the dimensionality of the observations $\mathbf{y}_t$.

In the above the updates are based on the entire sequence $\mathbf{y}_{1:T}$. We can neglect subsequences for which the probability of being in subtree $m$ is sufficiently low. This would result in similar updates as above, but with sums over only part of $1 : T$.

Missing values can be dealt with by treating missing (components of) $\mathbf{y}_t$'s as latent variables and integrating them out. In Algorithms 1 and 2 only the potentials change. In the M-step only the updates for $r_j^2$ and $C_j$ change. The noise covariance $r_j^2$ is effectively updated on less (weighted) data points, the projection matrix $C_j$ is updated one row at a time.

## References


[1] Michael. E. Tipping and Christopher. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, no. 3, 1999.

[2] Sam Roweis, "EM algorithms for PCA and SPCA," in *Neural Information Processing Systems 10*, 1997.

[3] Christopher M. Bishop and Michael. E. Tipping, "A hierarchical latent variable model for data visualization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, 1998.

[4] Uri Lerner and Ronald Parr, "Inference in hybrid networks: Theoretical limits and practical algorithms," in *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2001)*. 2001, Morgan Kaufmann Publishers.

[5] Thomas Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2001)*. 2001, Morgan Kaufmann Publishers.

[6] P.J. Harrison and C.F. Stevens, "Bayesian forecasting," *Journal of the Royal Statistical Society Society B*, vol. 38, pp. 205–247, 1976.

[7] Yaakov Bar-Shalom and Xiao-Rong Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, 1993.

[8] Joe Whittaker, *Graphical Models in Applied Multivariate Statistics*, John Wiley & Sons, 1989.

[9] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[10] T. Minka, "The EP energy function and minimization schemes," Tech. Rep., MIT Media Lab, 2001.

[11] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," in *NIPS 13*, 2001, pp. 689–695.

[12] Tom Heskes and Onno Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2002)*, San Francisco, CA, 2002, Morgan Kaufmann Publishers.

[13] Y. Teh and M. Welling, "The unified propagation and scaling algorithm," in *Advances in Neural Information Processing Systems 14*. 2002, p. (in press), MIT Press.

[14] Esa Alhoniemi, Jaakko Hollmén, Olli Simula, and Juha Vesanto, "Process monitoring and modeling using the self-organizing map," *Integrated Computer Aided Engineering*, vol. 6, no. 1, 1999.

[15] Christopher M. Bishop, Geoff E. Hinton, and Iain G.D. Strachan, "GTM through time," in *IEE International Conference on Artificial Neural Networks*, Cambridge, 1997, pp. 111–116.

[16] P Tino and I Nabney, "Hierarchical gtm: constructing localized non-linear projection manifolds in a principled way," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 639–656, 2002.

[17] R.H. Shumway and D.S. Stoffer, "Dynamic linear models with switching," *Journal of the American Statistical Association*, vol. 86, pp. 763–769, 1991.

[18] C.-J. Kim and Ch. R. Nelson, *State-Space Models with Regime Switching*, MIT Press, 1999.

[19] C. Carter and R. Kohn, "Markov chain Monte Carlo in conditionally Gaussian state space models," *Biometrika*, vol. 83, no. 3, pp. 589–601, 1996.

[20] Arnoud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russel, "Rao-Blackwellized particle filtering for dynamic Bayesian networks," in *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2000)*, San Francisco, CA, 2000, Morgan Kaufmann Publishers.

[21] Sam Roweis, Lawrence Saul, and Geoff Hinton, "Global coordination of local linear models," in *Neural Information Processing Systems 14*, 2001.

[22] Matthew Beal and Zoubin Ghahramani, "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems 13*, 2001.

[23] Sylvia Frühwirth-Schnatter, "Fully Bayesian analysis of switching Gaussian state space models," *Annals of the Institute of Statistical Mathematics*, vol. 53, no. 1, pp. 31–49, 2001.

[24] T. Heskes and O. Zoeter, "Generalized belief propagation for approximate inference in hybrid Bayesian networks," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, C. M. Bishop and B. J. Frey, Eds., 2003.

[25] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, Oxford University Press, 2001.