Variational inference for latent nonlinear dynamics

Daniel Hernandez
Department of Statistics
Columbia University

Liam Paninski
Department of Statistics
Columbia University

John Cunningham
Deparment of Statistics
Columbia University

Abstract

We introduce Variational Inference for Nonlinear Dynamics, an inference framework in the spirit of the Kalman filter, for learning a wide class of latent variable models with nonlinear hidden dynamics. The framework includes a novel algorithm, based on the fixed-point iteration method. We illustrate the technique in a model representing locally linear evolution with Gaussian or Poisson observations. It is shown that in this case, it outperforms state-of-the-art methods.

1 Introduction

Given a set of correlated observations $\mathbf{X} \equiv \{\mathbf{x}_1, \ldots \mathbf{x}_T\}$, $\mathbf{x}_t \in \mathbb{R}^{d_X}$, a latent variable model posits an underlying set of hidden variables $\mathbf{Z} \equiv \{\mathbf{z}_1, \ldots \mathbf{z}_T\}$, $\mathbf{z}_t \in \mathbb{R}^{d_Z}$, evolving through a dynamical law. In recent times, much work has been devoted to the inference problem of finding the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ [1, 2, 3, 5, 6, 7, 8, 9]. This problem is in general analytically intractable.

Here we present Variational Inference for Nonlinear Dynamics (VIND), a novel variational framework that is able to infer an approximate posterior representing nonlinear evolution in the latent space. Observations are expressed as a noise model, Poisson or Gaussian, operating on arbitrary smooth nonlinear functions of the latent state. The algorithm draws both from recent advances in variational inference [1, 2] and stochastic gradient variational Bayes [10, 11, 12], and makes use of a novel technique based on the fixed-point iteration method in order to find an approximate posterior describing nonlinear dynamics.

We use VIND to develop inference for a Locally Linear Dynamical System (LLDS) from observations specified by i) an in-model generated Poisson process and ii) Gaussian observations on top of the Lorenz system. We compare its performance with fLDS[1] and show that it significantly outperforms it, particularly in its accuracy to infer the hidden dynamics.

2 Theory

The VIND framework is suited for inference in continuous latent variable models. The starting point is the joint distribution $p_\varphi(\mathbf{X}, \mathbf{Z})$. The ELBO bound for the marginal loglikelihood $\log p_\varphi(\mathbf{X})$ with respect to a proposal posterior distribution $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ is given by:

$$\log p_\varphi(\mathbf{X}) \geq \mathscr{L}(\mathbf{X}) = \mathop{\mathbb{E}}_q[\log p_\varphi(\mathbf{X}, \mathbf{Z})] - \mathop{\mathbb{E}}_q[\log q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})] \,. \tag{1}$$

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Note that in general, $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ may share some parameters φ with the generative model. The joint $p_\varphi(\mathbf{X},\mathbf{Z})$ in Eq. (1) has the form:

$$p_\varphi(\mathbf{X},\mathbf{Z}) = c_p(\varphi) \cdot f_0(\mathbf{z}_0) \prod_{t=0}^{T} f_\varphi(\mathbf{x}_t|\mathbf{z}_t) \cdot H_\varphi(\mathbf{Z}) \,. \tag{2}$$

Here, f_0 is a prior, $f_0 \sim \mathcal{N}(\mu_0, \sigma_0)$, and the unnormalized densities f_φ stand for an observation model: $\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{P}\big(\lambda(\mathbf{z}_t)\big)$; where $\mathcal{P}(\lambda)$ is a noise model, parameterized by λ. For our purposes, $\mathcal{P}$ is taken to be either Poisson or Gaussian. The density $H_\varphi(\mathbf{Z})$ in Eq. (2) represents a model for the evolution of the latent variables. Note that, although it is assumed that a closed expression for the posterior $p_\varphi(\mathbf{Z}|\mathbf{X})$ cannot be obtained, it is known that it factorizes as,

$$p_\varphi(\mathbf{Z}|\mathbf{X}) = \frac{F(\mathbf{Z},\mathbf{X})}{p_\varphi(\mathbf{X})} \cdot H_\varphi(\mathbf{Z}) \,, \tag{3}$$

where F represents all the $\mathbf{X}$-dependent terms in Eq.(2).

The idea of VIND is to craft the variational approximation $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ in Eq. (1) to profit from what is known of $p_\varphi(\mathbf{Z}|\mathbf{X})$ from Eq. (3). To do so, define a *parent* posterior density $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ as:

$$Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) \propto h_0(\mathbf{z}_0) \prod_{t=0}^{T} g_\phi(\mathbf{z}_t|\mathbf{x}_t) \cdot H_\varphi(\mathbf{Z}) \,. \tag{4}$$

where $h_0(\mathbf{z}_0)$ is prior, $g_\phi(\mathbf{z}_t|\mathbf{x}_t) \sim \mathcal{N}\big(\mathbf{m}_\phi(\mathbf{x}_t), \mathbf{d}_\phi(\mathbf{x}_t)\big)$ - see [2] - and note that $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ includes explicitly the generative evolution term $H_\varphi(\mathbf{Z})$. VIND's prescription is to choose $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ as a Laplace approximation to $Q_{\phi,\varphi}$. That is, $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ is normal, with $\mathbf{X}$-dependent mean and variance, $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) \sim \mathcal{N}\big(\mathbf{P}_{\phi,\varphi}(\mathbf{X}), \mathbf{\Sigma}_{\phi,\varphi}(\mathbf{X})\big)$, and $\mathbf{P}_{\phi,\varphi}(\mathbf{X})$ is a mode of $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$. In specific cases, the fLDS model of [1, 2] for example, the functions $\mathbf{P}_{\phi,\varphi}$ and $\mathbf{\Sigma}_{\phi,\varphi}$ can be computed in closed form. This will not be possible in general. However, for a large class of models, the mean of the Laplace approximation appears as the solution to an equation that can be written in the form:

$$\mathbf{P} = r_{\phi,\varphi}(\mathbf{X},\, \mathbf{P}) \,. \tag{5}$$

where $r_{\phi,\varphi} \in \mathbb{R}^{T \times d_Z}$. For instance, a sufficient condition for Eq. (5) to hold is that $\log H_\varphi(\mathbf{Z})$ contains quadratic terms on each $\mathbf{z}_t$. In that case, the Laplace mean and variance can be obtained through a fixed-point iteration:

$$\mathbf{P}_{\phi,\varphi}(\mathbf{X}) = r_{\phi,\varphi}(\mathbf{X},\, r_{\phi,\varphi}(...,\, r_{\phi,\varphi}(\mathbf{X}, P_0))\dots))\,, \quad \text{iterate } m \text{ steps until convergence} \tag{6}$$

$$\mathbf{\Sigma}_{\phi,\varphi}(\mathbf{X}) = s_{\phi,\varphi}(\mathbf{X},\, \mathbf{P}_{\phi,\varphi}(\mathbf{X})) \tag{7}$$

where $s_{\phi,\varphi} \in \mathbb{R}^{T \times d_Z^2}$ and $P_0 \in \mathbb{R}^{T \times d_Z}$ is an initial path in $\mathbf{Z}$-space.

In order to use Eq. (6) in an optimization algorithm, an estimate for m is required. An improvement is to take a single iteration in Eq. (6) per epoch. Instead of waiting until convergence, we content ourselves with taking a step in the "right" direction,

$$\mathbf{P}^{(\text{ep})}_{\phi,\varphi}(\mathbf{X}) = r_{\phi,\varphi}\left(\mathbf{X},\, P^{(\text{ep})}\right)\,, \quad \mathbf{\Sigma}^{(\text{ep})}_{\phi,\varphi}(\mathbf{X}) = s_{\phi,\varphi}\left(\mathbf{X},\, P^{(\text{ep})}\right)\,, \quad \text{at epoch ep}\,. \tag{8}$$

The numeric mean posterior path estimate $P^{(\text{ep})}$ can be rechosen at each epoch. A natural choice is simply to take:

$$P^{(\text{ep})} = r_{\phi^{(\text{ep})},\varphi^{(\text{ep})}}\left(\mathbf{X},\, P^{(\text{ep}-1)}\right)\,, \tag{9}$$

where the numeric values $\phi^{(\text{ep})}$, $\varphi^{(\text{ep})}$ for the parameters at the current epoch are used in the RHS. The details of the algorithm are described in App. A.

VIND/LLDS

We now derive the functions $r_{\phi,\varphi},\ s_{\phi,\varphi}$ for the case in which $H_\varphi(\mathbf{Z})$ represents locally linear evolution. That is, we consider here an evolution model of the form:

$$H_\varphi(\mathbf{Z}) = \prod_{t=1}^{T} h_\varphi(\mathbf{z}_{t+1}|\mathbf{z}_t) \tag{10}$$

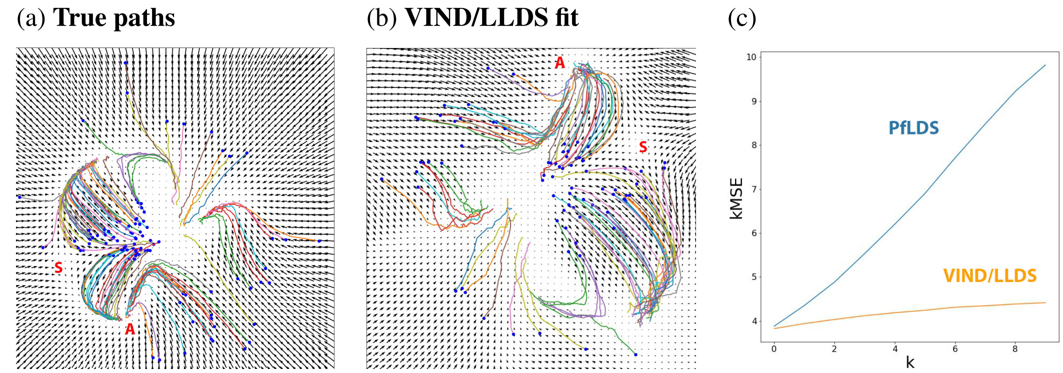

Figure 1: Inference of the latent dynamics for the Poisson in-model data. A trial corresponds to paths of the same color: (a) True paths state-space and dynamical system; (b) VIND/LLDS fit. Note how the underlying flow is correctly inferred, including the saddle point **S** and the attractor **A**. c) kMSE comparison between the Poisson fLDS [1] and the VIND/LLDS fits to this data.

where

$$h_\varphi(\mathbf{z}_{t+1}|\mathbf{z}_t) = \exp\left\{-\frac{1}{2}\big(\mathbf{z}_{t+1} - A_\varphi(\mathbf{z}_t)\mathbf{z}_t\big)^T \Lambda\big(\mathbf{z}_{t+1} - A_\varphi(\mathbf{z}_t)\mathbf{z}_t\big)\right\} . \tag{11}$$

Though $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ is itself intractable, a Laplace approximation can be computed. Write,

$$\log Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}) = \log c_Q - \frac{1}{2}\left[(\mathbf{Z}-\mathbf{M})^T\mathbf{R}_\phi(\mathbf{Z}-\mathbf{M}) + \mathbf{Z}^T\mathbf{S}_\varphi(\mathbf{Z})\mathbf{Z}\right] \tag{12}$$

where $\mathbf{R}_\phi$, $\mathbf{S}_\varphi(\mathbf{Z})$ are the precisions coming from the g_ϕ- and h_φ-terms respectively and $\mathbf{M} = \{\mathbf{m}_\phi(\mathbf{x}_1), \ldots, \mathbf{m}_\phi(\mathbf{x}_T)\}$. Setting the derivative w.r.t. $\mathbf{Z}$ to zero yields the fixed-point iteration equation for the mean $\mathbf{P}$, Eq. (5), with $r_{\phi,\varphi}$ given by,

$$r_{\phi,\varphi}(\mathbf{X},\mathbf{P}) = \left[\mathbf{R}_\phi + \mathbf{S}_\varphi(\mathbf{P})\right]^{-1}\left(\mathbf{R}_\phi\mathbf{M} - \frac{1}{2}\mathbf{P}^T\frac{\partial\mathbf{S}_\varphi(\mathbf{P})}{\partial\mathbf{P}}\mathbf{P}\right) \tag{13}$$

while $s_{\phi,\varphi}$ can be found by taking the second derivative of $\log Q_{\phi,\varphi}$ and replacing $r_{\phi,\varphi}$ in it.

3 Results

We compared how VIND/LLDS fares with respect to fLDS[1, 2] which fits linear dynamics. This comparison was performed across two datasets. First, we randomly generated an LLDS parameterized by a random $A_\varphi(\mathbf{z})$ and used it to generate a Poisson time series of counts per time bin. The second dataset consists of Gaussian observations on top of the classic Lorenz system. We describe below the results for these two datasets.

Our main quantitative measure of the quality of the inferred flow is the k-steps ahead mean squared error (kMSE) on test data. This measure is of particular value because, as opposed to the MSE which may be assessing only the merit of the autoencoder, kMSE evaluates the quality of the inferred dynamics. For both fLDS and VIND/LLDS, kMSE is computed as follows:

$$k\text{MSE} = \mathbb{E}\left[\sum_{t=0}^{T-k}(\mathbf{X}_{t+k} - \hat{\mathbf{X}}_{t+k})^2\right] , \tag{14}$$

where $\hat{\mathbf{X}}_{t+k}$ is the prediction at time $t+k$ and is obtained from $\mathbf{X}$ by first infering the value of $\mathbf{z}_t$, then applying $\mathbf{z}_{t+1} = A_\varphi(\mathbf{z}_t)\mathbf{z}_t$, k times, and finally using the generative model to compute the predicted observation at $t+k$, $\hat{\mathbf{X}}_{t+k}(\mathbf{Z}_{t+k})$. In Eq. (14), the expectation value is taken over trials and the dimension of the observations.

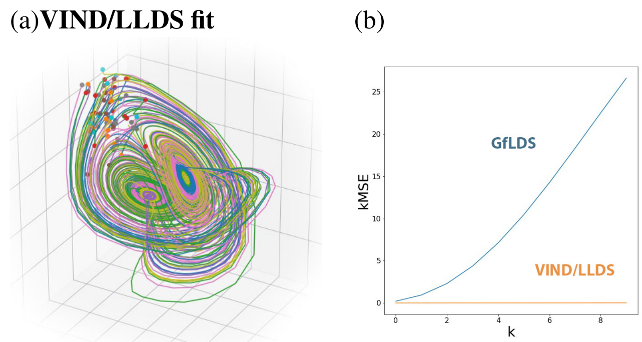

Figure 2: Results of the fit to the Lorenz DS with Gaussian observations: (a) Latent paths inferred by VIND/LLDS. The Lorenz attractor is clearly recognizable. (b) The kMSE for Gaussian fLDS [1](blue) and VIND/LLDS (orange); at 10 steps ahead, the difference is 3 orders of magnitude in this highly nonlinear system.

Poisson in-model data

We generated latent data using a nonlinear dynamical system defined by:

$$A_\varphi(\mathbf{z}) = f(|\mathbf{z}|)\big(\mathbb{I} + \alpha \cdot B_\varphi(\mathbf{z})\big) + 0.9 \cdot \big(1 - f(|\mathbf{z}|)\big) \cdot \mathbb{I}\,, \qquad f(x) = \frac{1}{2}\Big(1 - \tanh\big(a * (x - x_0)\big)\Big) \tag{15}$$

with $B(\mathbf{z}) \in \mathbb{R}^{d_Z^2}$ parameterized by a neural network with random parameters φ. This generative process is the same as the VIND/LLDS model, in that sense, this dataset is in-model. The function $f(|\mathbf{z}|)$ represents an inward flow at infinity; it ensures that the latent trajectories do not blow up. After sample paths are simulated, Poisson rates are obtained thorugh an observational model $\mathcal{P}$ parameterized by a neural network. Counts are finally generated through a Poisson process. In our experiments we used observations of dimensionality 10, 50 and 100, for 30 time bins. The trials were divided into training, validation and test sets in proportion 4:1:1.

Fig. 1 shows an example fits from VIND/LLDS to this data. In Fig. 1a and 1b, it is apparent how the inferred dynamics accurately captures the original including typical nonlinear features such as saddle points and attractors. Fig. 1c shows a comparison between the kMSE obtained with fLDS and VIND/LLDS showing a clear outperformance of the latter.

Lorenz system

The Lorenz system is a well known nonlinear system in 3 independent variables. We demonstrate the applicability of our framework to Gaussian observations on top of the Lorenz system:

$$\dot{y}_1 = \sigma(y_2 - y_1)\,, \quad \dot{y}_2 = y_1(\rho - y_3) - y_2\,, \quad \dot{y}_3 = y_1 y_2 - \beta y_3\,. \tag{16}$$

The parameter values $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ were used. The equations were numerically integrated using the 'odeint' integrator in the scipy python library. After the paths were computed, Gaussian observations were generated with constant variance across t. Fig. 2a shows the VIND/LLDS inferred paths for this system. The Lorenz attractor is clearly visible. The kMSE comparison, shown in Fig. 2b, demonstrates an improvement of 3 orders of magnitude over fLDS.

4 Discussion

We developed the VIND framework to infer an approximate Gaussian variational posterior for large classes of generative models on continuous latent variables, that is a substantial generalization of [1, 2]. Other than the latter, the most closely related works are [7] and [8]. None of these include the VIND prescription for sharing the parameters of the generative and recognition models. Moreover, [7] only applies to models with causal structure while, although not elaborated in this note, our framework can be applied also to more general graphical models in the latent space. Compared to [8], our method cannot handle discrete latent variables. The continuous state-space however allows for the other key novelty of this work: the use of the Laplace approximation - fixed-point iteration combo. We applied this framework to the case of an LLDS evolution model and we showed that accurate descriptions of the underlying dynamics are obtained.

References

[1] Yuanjun Gao, Evan Archer, Liam Paninski, John Cunningham (2016) Linear dynamical neural population models through nonlinear embeddings. *Advances in Neural Information Processing Systems (NIPS)*

[2] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, Liam Paninski (2015) Black box variational inference for state space models. *International Conference on Learning Representations (ICLR)*, Workshops.

[3] David Sussillo, Rafal Jozefowicz, LF Abbott, Chethan Pandarinath (2016) LFADS-Latent Factor Analysis via Dynamical Systems. *arXiv preprint arXiv:1608.06315*.

[4] Chethan Pandarinath et al. (2017) Inferring single-trial neural population dynamics using sequential auto-encoders. *bioRxiv (2017): 152884*.

[5] Yuan Zhao, Il Memming Park (2017) Recursive Variational Bayesian Dual Estimation for Nonlinear Dynamics and Non-Gaussian Observations. *arXiv preprint arXiv:1707.09049*.

[6] Yuan Zhao, Il Memming Park (2016) Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *arXiv preprint arXiv:1604.03053*.

[7] Rahul G Krishnan, Uri Shalit, David Sontag (2015) Deep Kalman filters. *arXiv preprint arXiv:1511.05121*.

[8] Johnson, Matthew, et al. (2016) Composing graphical models with neural networks for structured representations and fast inference." *Advances in neural information processing systems (NIPS)*.

[9] Maximilian Karl, Maximilian Soelch, Justin Bayer, Patrick van der Smagt (2016) Deep variational Bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*.

[10] Diederik P Kingma, Max Welling (2013) Auto-Encoding Variational Bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*

[11] Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra (2014) Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning, 2014*

[12] M. Titsias and M. Lázaro-Gredilla (2014) Doubly stochastic variational bayes for non-conjugate inference. *International Conference on Machine Learning*.

[13] Vasile Berinde (2006) *Iterative Approximation of Fixed Points.* Springer-Verlag.

A Algorithm

Here we describe the VIND algorithm referenced in the main text, and provide pseudocode. Below, $\mathbf{P}_{\phi,\varphi}$ and $\boldsymbol{\Sigma}_{\phi,\varphi}$ represent the ϕ, φ-dependent mean and variance of $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ while $P^{(\text{ep})}$ represents a numeric path in latent space, obtained by replacing ϕ and φ by its current values in Eq. (9).

Each epoch consists of two main steps: a gradient step, taken with respect to all the in-model variables, and an iteration step, as per Eq. (9), that updates $\mathbf{P}_{\phi,\varphi}$ and $\boldsymbol{\Sigma}_{\phi,\varphi}$ to approach the mean and variance of the parent distribution $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$. In order to compute an estimator for the gradient of the cost, Eq. (1), samples are extracted from $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ using the so called "reparameterization trick" [10, 11]

$$Z_i = \mathbf{P}_{\phi,\varphi}\big(\mathbf{X}, P^{(\text{ep})}\big) + \mathbf{O}_{\phi,\varphi}\big(\mathbf{X}, P^{(\text{ep})}\big)\epsilon_i \tag{17}$$

where $\epsilon_i \sim \mathcal{N}(0, \mathbb{I})$ and $\mathbf{O}_{\phi,\varphi}\mathbf{O}^T_{\phi,\varphi} = \boldsymbol{\Sigma}_{\phi,\varphi}$. Note that for the specific VIND/LLDS case, $\boldsymbol{\Sigma}^{-1}_{\phi,\varphi}$ is block-tridiagonal and the Cholesky decomposition can be performed efficiently, [2].

Precondition: $\mathbf{X}_i$ for $i = 1 \dots N$, N tensors of $T \times d_X$ observations of a time sequence with T time bins

ep $\leftarrow 0$. Initialize $\phi^{(\text{ep})}$, $\varphi^{(\text{ep})}$, $P^{(\text{ep})}$.
$\mathbf{P}^{(\text{ep})}_{\phi,\varphi} \leftarrow r_{\phi,\varphi}\big(\mathbf{X}, P^{(\text{ep})}\big)$, similarly $\boldsymbol{\Sigma}^{(\text{ep})}_{\phi,\varphi}$.
while not converged:
    for $i = 1$ to N:
        Get minibatch $\mathbf{X}_i$.
        Sample from $q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X}_i)$: $Z_i \sim \mathcal{N}(\mathbf{P}^{(\text{ep})}_{\phi,\varphi}, \boldsymbol{\Sigma}^{(\text{ep})}_{\phi,\varphi})$.
        Update ϕ, φ through $\nabla_{\phi,\varphi}\mathscr{L}_{\phi,\varphi}(\mathbf{X}_i, Z_i)$.
    ep $\leftarrow$ ep $+ 1$
    $P^{(\text{ep})} \leftarrow \mathbf{P}^{(\text{ep-1})}_{\phi^{(\text{ep})},\varphi^{(\text{ep})}}\big(\mathbf{X}\big)$. ▷ Evaluate ϕ, φ to get numeric $P^{(\text{ep})}$
    $\mathbf{P}^{(\text{ep})}_{\phi,\varphi} \leftarrow \mathrm{r}_{\phi,\varphi}\big(\mathbf{X}, P^{(\text{ep})}\big)$, similarly $\boldsymbol{\Sigma}^{(\text{ep})}_{\phi,\varphi}(\mathbf{X})$.

B Generic Markov Chain

Although we tested the VIND framework and algorithm for the specific choice of LLDS as the evolution model, VIND's applicability is not restricted to any particular dynamics. Instead, different choices of H_φ lead to correspondingly different $r_{\phi,\varphi}$ for use in the algorithm. To provide an example of this flexibility, we give here the recurrent equations for the type of models considered in [7]. In this case, the latent space evolution is generated via a Markov process, $\mathbf{z}_{t+1} \sim \mathcal{P}\big(T_\varphi(\mathbf{z}_t)\big)$ where $\mathcal{P}$ is a noise model. Consider then $p_\varphi(\mathbf{X}, \mathbf{Z})$ and $Q_{\phi,\varphi}(\mathbf{Z}|\mathbf{X})$ as above, with H_φ defined as in Eq. (10), and take h_φ as:

$$h_\varphi(\mathbf{z}_{t+1}|\mathbf{z}_t) = \exp\left\{-\frac{1}{2}\big(\mathbf{z}_{t+1} - T_\varphi(\mathbf{z}_t)\big)^T \Lambda \big(\mathbf{z}_{t+1} - T_\varphi(\mathbf{z}_t)\big)\right\} . \tag{18}$$

That is, now T_φ is a generic nonlinear function of $\mathbf{z}_t$. In this case, we can again obtain an implicit equation for the Laplace approximation of $Q_{\phi,\varphi}$,

$$\mathbf{z}_0 = \mathbf{m}(\mathbf{x}_0) + \mathbf{d}(\mathbf{x}_0)^{-1}\Big(\partial_{\mathbf{z}_0}(T_\varphi(\mathbf{z}_0)^T) \cdot \Lambda \cdot \big(\mathbf{z}_1 - T_\varphi(\mathbf{z}_0)\big)\Big) \tag{19}$$

$$\mathbf{z}_t = [\mathbf{d}(\mathbf{x}_t) + \Lambda]^{-1}\Big(\mathbf{d}_t\mathbf{m}_t + \Lambda T_\varphi(\mathbf{z}_{t-1}) + \partial_{\mathbf{z}_t}(T_\varphi(\mathbf{z}_t)^T) \cdot \Lambda \cdot \big(\mathbf{z}_{t+1} - T_\varphi(\mathbf{z}_t)\big)\Big) \tag{20}$$

$$\mathbf{z}_T = [\mathbf{d}(\mathbf{x}_T) + \Lambda]^{-1}\Big(\mathbf{d}_T\mathbf{m}_T + \Lambda T_\varphi(\mathbf{z}_{T-1})\Big) \tag{21}$$

which altogether define the function $r_{\phi,\varphi}$.