Revisiting the Structured Variational Autoencoder

Yixiu Zhao
Applied Physics Department
Stanford University
Stanford, CA, 94305
yixiuz@stanford.edu

Scott W. Linderman
Department of Statistics
Stanford University
Stanford, CA 94305
scott.linderman@stanford.edu

Abstract

The Structured Variational Autoencoder (SVAE) was introduced five years ago [Johnson et al., 2016]. It presented a modeling idea—to use probabilsitic graphical models (PGMs) as priors on latent variables and deep neural networks (DNNs) to map them to observed data—as well as an inference idea—to have the recognition network output conjugate potentials to the PGM prior rather than a full posterior. While mathematically appealing, the SVAE proved impractical to use or extend, as learning required implicit differentiation of a PGM inference algorithm, and the original authors' implementation was in pure Python with no GPU or TPU support. Now, armed with the power of JAX [Bradbury et al., 2018], a software library for automatic differentiation and compilation to CPU, GPU, or TPU targets, we revisit the SVAE. We develop a modular implementation that is orders of magnitude faster than the original code and show examples in a variety of different settings, including a scientific application to animal behavior modeling. Furthermore, we extend the original model by incorporating interior potentials, which allows for more expressive PGM priors, such as the Recurrent Switching Linear Dynamical System (rSLDS). Our JAX implementation of the SVAE and its extensions open up avenues for many practical applications, extensions, and theoretical investigations.

1 Introduction

Deep neural networks (DNNs) are the main workforce of artificial intelligence nowadays, due to their flexibility and scalability in representing complex high dimensional distributions. Another key factor in the popularity of deep neural nets is the (unreasonable) effectiveness of generic stochastic gradient-based optimization techniques. However, generic DNNs are hard to interpret, and quantifiably building prior knowledge into these architectures can be more of an art than science. On the other hand, probabilistic graphical models (PGMs) are often interpretable by design, which in turn allows for specialized algorithms for efficient inference. Furthermore, it is easy to design these graphical models to reflect our understanding of the underlying structure of the problem, and use conjugate priors to quantitatively reflect our prior knowledge. However, PGMs are not applicable to many real world problems because of their limited scalability and efficiency in performing inference in high dimensional settings. Therefore, many methods have combined deep neural networks and PGMs to leverage the best of both [Johnson et al., 2016, Archer et al., 2015, Krishnan et al., 2015, Lin et al., 2018, e.g.]. We revisit one such method—the Structured Variational Autoencoder Johnson et al. [2016]—and provide a modular implementation in JAX that is easy to use and extend. In this paper, we provide examples of different SVAE models implemented with our JAX code, as well as an extension to the original framework that enables a broader class of PGM priors, like recurrent switching linear dynamical systems [Linderman et al., 2017].

Bayesian Deep Learning workshop, NeurIPS 2021.

2 Structured Variational Autoencoders

Structured variational autoencoders are a special case of VAEs. Let $y \in \mathbb{R}^N$ denote an observed data point and $x \in \mathbb{R}^D$ be its latent representation. In an SVAE, the joint distribution factors into a prior $p(x; \theta)$ and a likelihood $p(y \mid x; \gamma)$, but rather than being a simple Gaussian, the prior is allowed to be any PGM built of conjugate, exponential family conditional distributions with parameters θ. The likelihood is assumed to be nonconjugate with the prior, and it is usually implemented by a deep neural network with weights γ.

Learning and inference in the SVAE are done by stochastic gradient ascent on the evidence lower bound (ELBO),

$$\mathcal{L}(\phi, \theta, \gamma) = \mathbb{E}_{q(x;y,\theta,\phi)}[\log p(x, y; \theta, \gamma) - \log q(x; y, \theta, \phi)] \leq \log p(y \mid \theta, \gamma), \tag{1}$$

where ϕ are parameters of the recognition network. The key inference idea is to have the recognition network output *conjugate potentials* $\psi(x; y, \phi)$ that is linear in the sufficient statistics of the exponential family prior. Together, the prior and conjugate potentials define a surrogate model,

$$\tilde{p}(x; y, \theta, \phi) \propto p(x; \theta) \, \exp\{\psi(x; y, \phi)\}. \tag{2}$$

The variational posterior is then implicitly defined as the solution to the surrogate variational inference problem:

$$q(x; y, \theta, \phi) = \arg\min_{\tilde{q} \in Q} \mathrm{KL}(\tilde{q}(x) \,||\, \tilde{p}(x; y, \theta, \phi)) \tag{3}$$

where Q is the variational family. Since the potentials are conjugate with the PGM prior, solving the surrogate variational inference problem often admits existing inference algorithms. For example, when Q is the a mean-field variational family, we can perform coordinate ascent variational inference (CAVI) to find a local optimum q. We can view the conjugate potential $\psi(x; y)$ as approximations of the log likelihood function $\log p(y \mid x)$ that are linear in the sufficient statistics of the prior. For example, if the prior were a multivariate Gaussian with sufficient statistics $t(x) = (x, xx^\top)$, the conjugate potentials would be of the form $\psi(x; y, \phi) = \langle h(y, \phi), x \rangle + \langle J(y, \phi), xx^\top \rangle$; i.e. a quadratic approximation to the log likelihood.

The challenge is that taking gradients of the ELBO with respect to the recognition parameters ϕ requires differentiating the implicitly defined posterior density in eq. (3). Since the SVAE was first introduced, implicit models have become much more widely used [Duvenaud et al., 2020], fueled in part by better software tools like PyTorch and JAX. We developed a JAX implementation that makes SVAEs practically useful[1], we demonstrate it on a variety of test problems, and show an extension to a model where the prior is only partially conjugate.

3 Examples

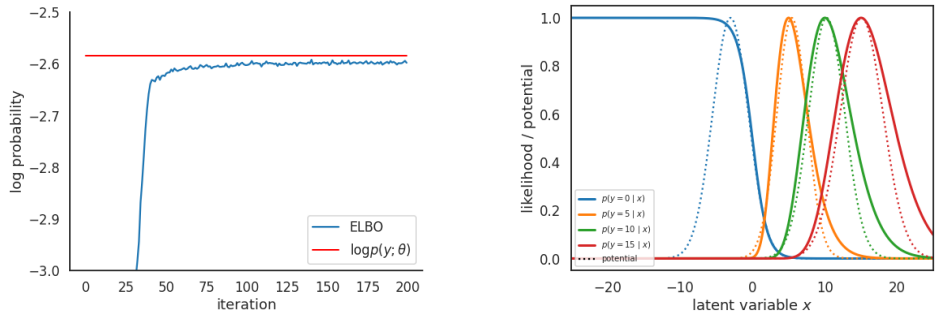
The Gaussian Poisson model First we consider a very simple toy example:

$$x_n \sim \mathcal{N}(\mu, \sigma), \; y_n \sim \mathrm{Po}(\mathrm{Softplus}(x_n))$$

Here $x_n \in \mathbb{R}$ is a real-valued latent variable for the n-th datapoint, and $y_n \in \mathbb{N}$ is the count-valued observation. We sampled 10K datapoints from this generative model and fit our model with a fixed prior for simplicity. The SVAE converges at around 100 training iterations (figure 1a). It is worth noting that this simple model demonstrates an important limitation of the SVAE: the Poisson likelihood function is not well approximated by Gaussian potentials, as shown in figure 1b. This means that there will always be an approximation gap between the model ELBO and the true log likelihood of the data (the red line in figure 1a). This approximation gap motivates the use of a more flexible class of potentials, such as mixtures of Gaussians, as one possible extension of the framework.

Mouse behavioral segmentation with switching linear dynamical systems The switching linear dynamical system (SLDS) is a popular probabilistic model for segmenting complex systems with nonlinear dynamics into simple linear systems. Here we use an SLDS-SVAE to model mouse behavior

[1] Check out our Jupyter notebook at `t.ly/a8GV`



(a) (b)

Figure 1: The Gaussian Poisson model. (a) the training curve over 200 iterations. Note the gap between obtained ELBO and the true data log likelihood. (b) the ELBO gap is caused by the Gaussian potential's inability to approximate Poisson likelihoods accurately.

videos in order to discover interpretable behavior syllables of free roaming mice Wiltschko et al. [2015]. The probabilistic model is as follows:

$$z_0 \sim \text{Cat}(\pi_0), \qquad z_t|z_{t-1} \sim \text{Cat}(\pi_{z_{t-1}}),$$
$$x_0 \sim \mathcal{N}(\mu_{z_0}, \Sigma_{z_0}), \quad x_t|x_{t-1} \sim \mathcal{N}(A_{z_t}x_{t-1} + b_{z_t}, \Sigma_{z_t}),$$

where $z_{1:T}$ and $x_{1:T}$ are the discrete and continuous latent states respectively. We use convolution architectures similar to the ones in [Batty et al., 2019] for the recognition and generator networks. We fit the model to mouse behavior videos in Wiltschko et al. [2015] and visualize the learned representations and discrete states (the "syllables" of behavior) in figures 2a and 2b. We see that the model learns to categorize typical behaviors of free roaming mice including darting, rearing, etc.

Recurrent switching linear dynamical system One downfall of the SLDS model, as pointed out in Linderman et al. [2017] is that the discrete state is fully autonomous, while in the real world many dynamical systems change their states based on where they are in the state space. For example, a ball bouncing inside a box will only enter the "in collision" state when it touches one of the edges of the box. This motivates the recurrent switching linear dynamical system (rSLDS), where an additional dependence of the discrete state transition on the continuous state is added, changing the discrete state transition to:

$$z_t|z_{t-1}, x_{t-1} \sim \text{Cat}(\pi(z_{t-1}, x_{t-1})), \ \pi(z_t, x_t) = \text{Softmax}(Wx_t + \pi_{z_t} + c)$$

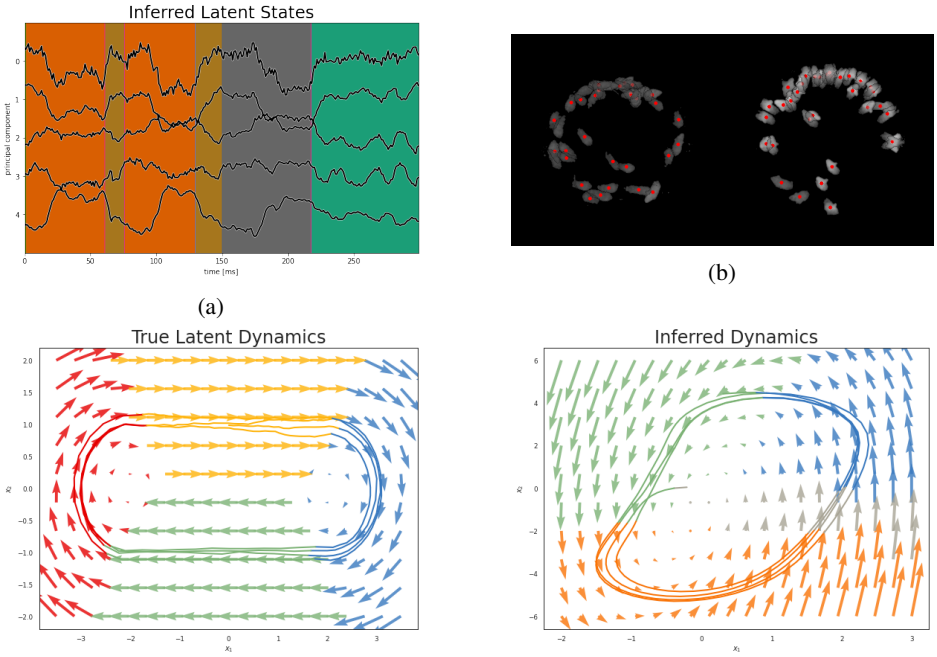
where W, c are parameters of the affine mapping. Variational inference in this model is tricky since the additional arrows from x_{t-1} to z_t breaks the conjugacy structure. To accommodate this, we extend the SVAE framework by introducing interior dynamics potentials:

$$\psi_d(z_t, z_{t-1}, x_{t-1}) = \prod_{i,j}[C_{ij}\mathcal{N}(x_{t-1}|J_{ij}, h_{ij})]^{\mathbb{I}(z_t=i)\mathbb{I}(z_{t-1}=j)},$$

where J_{ij} and h_{ij} are the natural parameters of the Gaussian potentials, and C_{ij} is a scaling constant, thereby replacing non-conjugate terms with learnable conjugate terms. Similarly, in the prior parameter updates, we use gradient descent to update the softmax weights W and c since they are non-conjugate. We use the NASCAR® toy dataset Linderman et al. [2017] as a proof of concept, where the discrete transitions are entirely dependent on the continuous states. The SVAE model is successful in inferring the 2-dimensional latent dynamics (up to a linear transformation) as well as the prior dynamics from 5-dimensional observations, as shown in figure 2c, 2d. Although the learned prior dynamics are not perfect, the model captures the 4 discrete regimes reasonably well. More work remains to be done to scale this model to realistic datasets with high-dimensional observations.

4 Conclusion

The original SVAE is known by many to be hard to implement and train. With this JAX implementation of the framework, we hope to eliminate some of the obstacles to training and using SVAEs,



(a) (b) (c) (d)

Figure 2: Examples with switching linear dynamical system priors. (a) the 5-dimensional latent representation of mouse behavior inferred by the model. The colored regions represent discrete states. (b) two examples of the discrete states (behavior syllables) learned by the model. (c) the latent dynamics from which the data is generated. (d) the SVAE model is able to recover the dynamics and learn the discrete states reasonably well, without any knowledge of the prior parameters.

and make the model accessible to a wider audience. We hope that this work can inspire further investigations on how to train hybrid neural network and probabilistic models more efficiently and effectively, making viable a class of powerful models that gets the best of both worlds.

References

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29:2946–2954, 2016.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Wu Lin, Nicolas Hubacher, and Mohammad Emtiyaz Khan. Variational message passing with structured inference networks. *arXiv preprint arXiv:1803.05589*, 2018.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922. PMLR, 2017.

David Duvenaud, Zico Kolter, and Matthew Johnson, Dec 2020. URL `https://implicit-layers-tutorial.org/`.

Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.

Eleanor Batty, Matthew Whiteway, Shreya Saxena, Dan Biderman, Taiga Abe, Simon Musall, Winthrop Gillis, Jeffrey Markowitz, Anne Churchland, John Cunningham, et al. Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. 2019.