Variational Structured Stochastic Network

Hao Liu 1 Xinyi Yang 2 Zenglin Xu 2

Abstract

High dimensional sequential data exhibits complex structure, a successful generative model for such data must involve highly dependent, structured variables. Thus it is desired or even necessary to model correlations and dependencies between the multiple input, output variables and latent variables in such scenario. To achieve this goal, we introduce Variational Structured Stochastic Network (VSSN), a new method for modeling high dimensional structured data. Leveraging recent advances in Stochastic Gradient Variational Bayes, VSSN can overcome intractable inference distributions via stochastic variational inference (Hoffman et al., 2013; Ranganath et al., 2014). To evaluate the proposed model, we apply it to speech recording data, music data, and several dynamic image sequence modeling tasks. Experimental results have demonstrated that our proposed method can outperform most state-of-the-art methods.

1. Introduction

Learning structured generative models for high dimensional sequential data is a critical yet challenging research topic in machine learning (Eyjolfsdottir et al., 2016; Johnson et al., 2016; Graves, 2013; Watter et al., 2015; Chung et al., 2015). This problem has been studied for many decades using State Space Models (SSMs) such as Hidden Markov Models (HMMs) and Kalman filters (Roweis & Ghahramani, 1999). Another popular method is Recurrent Neural Network (RNN), a recurrent type of neural network, is employed to handle both of them for structured prediction (Johnson et al., 2016; Eyjolfsdottir et al., 2016; Fraccaro et al., 2016; Chung et al., 2015). However, for complicated input data, it is difficult for standard SSMs or RNNs to accurately model the underlying dependency structures (Chung et al., 2015; Gu et al., 2015; Gan et al., 2015; Sutskever et al., 2014). Therefore, it is essential to develop learning algorithm that with a good representation of the structure of input variables, output variables and latent states.

To address these issues, we propose an efficient and scalable model called Variational Structured Stochastic Network (VSSN), which enables encode long-term structured dependency in generative model by providing the corresponding inference mechanism with rich capacity.

2. Model

We start by specifying our generative and variational models and then introduce the proposed algorithm.

2.1. Generative model

Consider non-linear dynamical systems with observations \( x_t \in X \subset \mathbb{R}^{n_x} \), depending on control inputs \( u_t \in U \subset \mathbb{R}^{n_u} \). Elements of \( X \) can be high-dimensional sensory data, e.g., raw images. In particular they may exhibit complex non-Markovian transitions. Corresponding time-discrete sequences of length \( T \) are denoted as \( \mathbf{x}_{1:T} = (x_1, x_2, ..., x_T) \) and \( \mathbf{u}_{1:T} = (u_1, u_2, ..., u_T) \). We write VSSN as a generative model that temporally interlocks an SSM with a RNN, as illustrated in Figure 1 for a single sequence model. For a single sequence setting, we have

\[
L(\theta) = \log p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}, d_0, z_0),
\]

where \( \theta \) denotes the set of parameters, \( d_0 \) and \( z_0 \) are initial latent states. It is important to note that when there are \( N \) sequences in a dynamic system, we can write the likelihood of the \( i \)-th sequence, i.e., \( L_i(\theta) \), in the form of Equation 1 and formulate the whole likelihood as \( L(\theta) = \sum_{i=1}^{N} L_i(\theta) \). For convenience of presentation, throughout the paper, we omit the index \( i \) when only one sequence is referred to, or when it is clear from the context. We set a prior \( p(\beta) \) on \( \beta_{1:T} \) and write the joint probability of the observation
sequence and its latent states as follows:

\[
p_\theta(x_{1:T}, z_{1:T}, d_{1:T}, \beta_{1:T}|u_{1:T}, z_0, d_0) = p_{\theta_z}(x_{1:T}|z_{1:T}, d_{1:T}),
\]

\[
p_{\phi_z}(z_{1:T}|d_{1:T}, z_0) \cdot p_{\theta_u}(d_{1:T}|u_{1:T}, \beta_{1:T}, d_0) \cdot p(\beta_{1:T})
\]

\[= \prod_{t=1}^{T} p_{\theta_z}(x_t|z_t, d_t)p_{\phi_z}(z_t|z_{t-1}, d_t)p_{\theta_u}(d_t|d_{t-1}, \beta_t, u_t)p(\beta_t),
\]

where \(\theta_x\), \(\theta_z\), and \(\theta_{\beta}\) denote the parameters related to the corresponding conditional distributions. And we have \(\theta = \{\theta_x, \theta_z, \theta_{\beta}\}\).

The log likelihood term \(L(\theta)\) in Equation 1 could be calculated by averaging out the latent states \(z_{1:T}\) and \(d_{1:T}\) from Equation 2. Following Figure 1, the states \(d_{1:T}\) are determined from \(d_0, \beta_{1:T}\) and \(u_{1:T}\) through the recursion \(d_t = f_{\theta_d}(d_{t-1}, u_t, \beta_t)\). In our implementation \(f_{\theta_d}\) is a GRU network with parameters \(\theta_d\).

We assume \(p_{\phi_z}(z_t|z_{t-1}, d_t)\) is subject to a Gaussian distribution with a diagonal covariance structure, namely \(p_{\phi_z}(z_t|z_{t-1}, d_t) = \mathcal{N}(z_t; \mu_t, v_t)\). In particularly, its mean and log-variance are parameterized by neural networks depending on \(z_{t-1}\) and \(d_t\), as follows,

\[
\mu_t = f_1(z_{t-1}, d_t), \log v_t = f_2(z_{t-1}, d_t).
\]

where \(f_i(\cdot)\) for \(i = 1, 2\) denotes a neural network, respectively. We split \(\beta_t = (w_t, v_t)\), where \(w_t\) is the a sample-specific process noise which can be inferred from incoming data, while \(v_t\) are universal transition parameters, which are sample-independent (and are only inferred from data during training). Thus it leads to a decomposition as follows:

\[
q_{\phi_z}(\beta_{1:T}|x_{1:T}, u_{1:T}) = q_{\phi_z}(w_{1:T}|x_{1:T})q_{\phi_z}(v_{1:T}).
\]

### 2.2. Inference Network

Instead of maximizing \(L(x)\) with respect to \(\theta\), we maximize a variational evidence lower bound (ELBO) over \(L(x)\), i.e., \(F(\theta, \phi) \leq L(x)\), with respect to both \(\theta\) and the variational parameters \(\phi\).

\[
F(\theta, \phi) = \int \int q_{\phi}(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, S) \cdot \log \frac{p_{\theta}(x_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, S)}{q_{\phi}(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, S)} \, dd_{1:T}dz_{1:T}d\beta_{1:T},
\]

where \(S = \{u_{1:T}, d_0, z_0\}\) denote the set of fixed variables. Maximizing \(F(\theta, \phi)\) with parameters \(\theta\) and \(\phi\) is done by stochastic gradient ascent, and in doing so, both the posterior and its approximation \(q_{\phi}\) change simultaneously. In general, intractable expectations in the objective function can typically be approximated by the reparameterization trick (Kingma & Welling, 2013; Gu et al., 2015; Chung et al., 2016) or control variates (Paisley et al., 2012) to obtain low-variance estimators of its gradients. In order to obtain efficient solution for \(F(\theta, \phi)\), we adopt the reparameterization trick. We add initial structure to \(q_{\phi}\) by noticing that the prior \(p_{\phi}(z_{1:T}|d_{1:T}, z_0)\) in the generative model is a delta function over the computed \(z_{1:T}\), and so is the posterior \(p_{\phi}(z_{1:T}|x_{1:T}, d_{1:T}, z_0)\). Consequently, we let the inference network use exactly the same deterministic state setting \(z_{1:T}\) as that of the generative model, and we decompose it as:

\[
q_{\phi}(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, u_{1:T}, z_0) = q(z_{1:T}|x_{1:T}, z_0, d_{1:T}) \cdot q(d_{1:T}|u_{1:T}, x_{1:T}, \beta_{1:T}, d_0) \cdot q(\beta_{1:T}|x_{1:T}, u_{1:T}).
\]

Note that \(q(d_{1:T}|u_{1:T}, x_{1:T}, \beta_{1:T}, d_0)\) is exactly equals to \(p_{\phi}(d_{1:T}|u_{1:T}, \beta_{1:T}, d_0)\) based on the generative graphical model Figure 1. Then we substitute Eq. 3 into \(F(\theta, \phi)\) and
by some math manipulation:
\[ F(\theta, \phi) \geq \int \{ \int q_{\beta}(\beta|x_{1:T}, u_{1:T}) \log p_{\theta}(x_{1:T}|z_{1:T}, d_{1:T}) p(\beta|1, T) \\
q_{\phi_1}(z_{1:T}|d_{1:T}, x_{1:T}, z_0) q_{\phi_2}(\beta_1|x_{1:T}, u_{1:T}) \} d\beta \] 

Then we factorize \( p_{\theta}(x_{1:T}|z_{1:T}, d_{1:T}) \) to two terms \( \sqrt{p_{\theta}(x_{1:T}|z_{1:T}, d_{1:T})} \) under log and the bound in Eq.(4) can be further optimized as follow:

\[ \text{Eq. (4)} \geq \frac{1}{2} \int \{ E_{q_{\phi_1}}(z_{1:T}|d_{1:T}, x_{1:T}, z_0) \} \{ E_{q_{\phi_2}}(\beta_{1:T}|x_{1:T}, u_{1:T}) \} \]

\[ \log p_{\theta}(x_{1:T}|z_{1:T}, d_{1:T}) - KL(q_{\phi}(\beta_{1:T}|x_{1:T}, u_{1:T})||p(\beta_{1:T})) + \frac{1}{2} E_{q_{\phi_2}}(\beta_{1:T}|x_{1:T}, u_{1:T}) \}

\[ - \frac{1}{2} \log q_{\phi_1}(z_{1:T}|d_{1:T}, x_{1:T}, z_0) \] 

\[ \frac{1}{2} \int \log - \frac{1}{q_{\phi_1}(z_{1:T}|d_{1:T}, x_{1:T}, z_0)} dz := L(\theta, \phi) \]

Here KL denotes the Kullback-Leibler divergence between two distributions. The lower bound is denoted as L(\( \theta, \phi \)). We denote the first term in \( F(\theta, \phi) \) as A while the second term denoted as B for ease of simplification.

The true posterior distribution of the stochastic states \( h_{1:T} \), given both the data and the deterministic state \( z_{1:T} \), factorizes as:

\[ q_{h}(z_{1:T}|d_{1:T}, x_{1:T}, z_0) = \prod_t q_{h}(z_t|z_{t-1}, d_{t:T}, x_{t:T}) \]

This shows that, knowing \( z_{t-1} \), the posterior distribution of \( z_t \) does not depend on the past outputs, but only on the present and future ones; this was also noted in (Krishnan et al., 2015; 2016). Instead of factorizing \( q_{\phi} \) as a mean-field approximation across time steps, we keep the structured form of the posterior factors, including \( z_t \)s dependence on \( z_{t-1} \), in the variational approximation:

\[ q_{\phi_1}(z_{1:T}|d_{1:T}, x_{1:T}, z_0) = \prod_t q_{\phi_1}(z_{t}|z_{t-1}, d_{t:T}, x_{t:T}) \]

\[ = \prod_t q_{\phi_1}(z_{t}|z_{t-1}, a_t) \]

(4)

where \( a_t = g_{\phi_2}(a_{t+1}, [d_t, x_t]) \) and \( [d_t, x_t] \) is the concatenation of the vectors \( z_t \) and \( x_t \). We mimic each posterior factor’s nonlinear long-term dependence on \( d_{t:T} \) and \( x_{t:T} \) through a backwards-recurrent function \( g_{\phi_2} \). The inference network is therefore parameterized by \( \phi = \{ \phi_1, \phi_2, \phi_3 \} \). The inference procedure is illustrated in Figure 2.

As both the generative model and inference network factorize over time steps, the ELBO separates as a sum over the time steps:

\[ B = \sum_t E_{q_{\phi_1}}(z_{t-1}) \{ E_{q_\phi}(z_t|z_{t-1}, d_{t:T}, x_{t:T}) \} \log p_{\theta}(x_t|z_t, d_t) \]

\[ - KL(q_{\phi_1}(z_{t-1}, d_{t:T}, x_{t:T})||p_{\theta}(z_t|z_{t-1}, d_t)) \]

3. Experiment

3.1. Synthetic Experiment

To validate that VSSN is able to model high dimensional data with complex dependency, we simulated a dynamic torque-controlled pendulum governed by the differential equation to test VSSN on non-Markovian observations of a dynamical system: \( \mu \ddot{\phi}(t) = -\mu \dot{\phi}(t) + mgl \sin \phi(t) \). For fair comparison with (Karl et al., 2016), we set \( m = l = 1, \mu = 0.5, g = 9.81 \), via numerical integration, and then converted the ground-truth angle into an image observation. The one-dimensional control corresponds to angle acceleration. Angle and angular velocity fully describe the system. The OLS regression results are shown in Table 2. VSSN is clearly better than DVBFL-LL and DKF in predicting \( \sin \phi, \cos \phi \), and \( \phi \) . VSSN achieves a higher goodness-of-fit than other methods.

3.2. Speech modelling

We also evaluate VSSN on the modelling of speech data, i.e., Blizzard and TIMIT datasets. Here Blizzard is a dataset of 300 hours of English speech by a single female speaker and TIMIT is a dataset of 6300 English sentences read by 630 speakers. This is a challenging task since they have shown to be difficult to model without a good representation of the uncertainty in the latent states (Chung et al., 2015; Gu et al., 2015; Gan et al., 2015; Sutskever et al., 2014). We report the results in Table 3. It can be seen that VSSN performs slightly better than SRN(smooth+ResQ) and other methods on TIMIT, while outperforms current state-of-the-art methods on Blizzard by a large margin.

3.3. Bouncing Balls

The bouncing balls dataset is a common test set for models that generate high dimensional sequences. It consists
of simulations of three balls bouncing in a box. We followed standard procedure to create 4000 training videos and 200 testing videos (Sutskever et al., 2009; Gan et al., 2015) and used an additional 200 videos for validation. It features three ball rolling within a bounding box in a plane. If the ball hits the wall and another ball, it bounces off, so that the true dynamics are highly dependent on the current position and velocity of the ball. Each video is of length 100 and of resolution 30 × 30. As can be seen, the model is able both to represent the ball almost accurately and to make long-term predictions while modelling uncertainty. VSSN outperforms the Deep Temporal Sigmoid Belief Network (Gan et al., 2015), the recurrent temporal RBM (RTRBM) and the structured RTRBM (SRTRBM)(Mittelman et al., 2014). VSSN also compete favorably with the Predictive Generative Networks(PGN) (Lotter et al., 2015) although PGN is a rather complex deep neural network. Results shown in Table 2. An example of prediction sequence is shown in Figure 3.

### 3.4. Polyphonic Music

Additionally, we test VSSN for modelling sequences of polyphonic music, using the four data sets of MIDI songs introduced. Each data set contains more than 7 hours of polyphonic music of varying complexity: folk tunes (Nottingham data set), the four-part chorales by J. S. Bach (JSB chorales), orchestral music (MuseData) and classical piano music (Piano-midi.de). Table 4 compares the average log-likelihood on the test sets obtained with the models introduced in (Bayer & Osendorfer, 2014; Gan et al., 2015; Gu et al., 2015). It can be seen that VSSN performs slightly better or by a large margin than other methods.

### 4. Conclusion

To learn a generative model for high dimensional sequential data, Variational Structured Stochastic Network is introduced here, which is a novel model for learning a good representation of the structure of input, output and latent variables.
References


