Structured Recurrent Temporal Restricted Boltzmann Machines

Roni Mittelman[†] RMITTELM@UMICH.EDU
Benjamin Kuipers[†] KUIPERS@UMICH.EDU
Silvio Savarese[‡] SSILVIO@STANFORD.EDU
Honglak Lee[†] HONGLAK@UMICH.EDU
[†]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI
[‡]Computer Science Department, Stanford University, Stanford, CA

Abstract

The recurrent temporal restricted Boltzmann machine (RTRBM) is a probabilistic time-series model. The topology of the RTRBM graphical model, however, assumes full connectivity between all the pairs of visible units and hidden units, thereby ignoring the dependency structure within the observations. Learning this structure has the potential for not only improving the prediction performance, but also revealing important dependency patterns in the data. For example, given a meteorological dataset, we could identify regional weather patterns. In this work, we propose a new class of RTRBM, which we refer to as the structured RTRBM (SRTRBM), which explicitly uses a graph to model the dependency structure. Our technique is related to methods such as graphical lasso, which are used to learn the topology of Gaussian graphical models. We also develop a spike-and-slab version of the RTRBM, and combine it with the SRTRBM to learn dependency structures in datasets with real-valued observations. Our experimental results using synthetic and real datasets demonstrate that the SRTRBM can significantly improve the prediction performance of the RTRBM, particularly when the number of visible units is large and the size of the training set is small. It also reveals the dependency structures underlying our benchmark datasets.

1. Introduction

Discovering patterns and relationships in static and time-series data is an important theme in many machine learning problems, spanning different fields such as modeling of human activities in videos (Prabhakar et al., 2010), or inferring gene regulatory networks (Mohan et al., 2012). Gaussian graphical models have been used often to learn the structure of static data, by using a graph to describe the sparsity pattern of the inverse covariance matrix. The topology of the graph captures the conditional independence property between non-adjacent nodes (Lauritzen, 1996). The structure of a Gaussian graphical model can be learned using the graphical lasso (Banerjee et al., 2008; Friedman et al., 2008) approach, which uses sparsity promoting regularization to encourage the learning of sparsely connected graphs. The graphical lasso framework is particularly important for estimation of the inverse covariance matrix when the number of observations is smaller than the number of variables.

The Graphical lasso has also been used to model sequence data, using graphical models of autoregressive processes (Songsiri & Vandenberghe, 2010). Other time-series models include hidden Markov models (Rabiner & Juang, 1986) and dynamic Bayesian networks (Doshi et al., 2011). However, as was discussed in Taylor et al. (2011), these models have difficulties capturing long range temporal dependencies. Another class of time-series models, which are potentially better suited to capture long range dependencies, relies on the use of recurrent neural networks (RNNs) (Rumelhart et al., 1986; Martens & Sutskever, 2011; Boulanger-Lewandowski et al., 2012; Sutskever et al., 2013; Pascanu et al., 2013; Bengio et al., 2013), and variants of restricted Boltzmann machines (RBMs) (Taylor et al., 2011; Sutskever & Hinton, 2007; Sutskever et al., 2008). However, these models assume a full connectivity between all the pairs of visible and hidden units, which makes it difficult to identify important dependency structures between observations and hidden units from the learned model.

In this paper, we develop a new approach to learn dependency structures in time-series data, based on the recurrent temporal restricted Boltzmann machine (RTRBM) (Sutskever et al., 2008). The RTRBM can be viewed as a temporal stack of RBMs, where each RBM has a *contextual* hidden state that is received from the previous RBM

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

and is used to modulate its hidden units bias. In our model, we use a dependency graph to define a new energy function for the RTRBM, such that the graph topology is used to determine appropriate *masking* matrices for the RTRBM model parameters. By replacing the elements of the masking matrices with logistic functions, we are able to learn the structure of the graph. Furthermore, we propose a sparsity promoting regularization to promote learning of sparsely connected graphs.

In essence, our approach, which we refer to as the *structured RTRBM (SRTRBM)*, shares similar motivations to graphical lasso, and can reveal structure in time-series data. Similarly to graphical lasso, we show that learning the structure in the RTRBM can improve the prediction performance, particularly when the size of the observations vector is large, and the size of the training set is small. It can also reveal important dependency patterns in the data.

Another contribution of this work is to develop a temporal extension of the spike-and-slab RBM (ssRBM) (Courville et al., 2011a;b) for modeling of time-series data with real-valued observations. Previously, modeling of real-valued observations using the RTRBM was achieved using a Gaussian RBM (GRBM) (Hinton & Salakhutdinov, 2006), which is known to suffer from inadequately modeling the conditional covariance of the visible units (Ranzato & Hinton, 2010; Dahl. et al., 2010). The ssRBM associates a real-valued variable ("slab") with each binary hidden unit ("spike"), and unlike the GRBM it has a conditional distribution for the visible units which has a non-diagonal covariance. We propose the spike-and-slab formulation in our SRTRBM in order to learn the structure of datasets with real-valued observations. Our experimental results verify that the spike-and-slab version of our model can improve over the Gaussian RTRBM, and that learning the structure can reduce the prediction error. We are also able to learn the underlying structure of our benchmark datasets.

The rest of the paper is organized as follows. Section 2 provides the background about the RBM and the ssRBM, and in Section 3 we review the RTRBM and its learning algorithm. We develop the SRTRBM in Section 4 and the spike-and-slab-based RTRBM and SRTRBM in Section 5. In Section 6 we present the experimental results, and Section 7 concludes this paper.

2. Preliminaries

2.1. Restricted Boltzmann Machines

The RBM (Smolensky, 1987) is an undirected graphical model that represents a joint distribution over visible units v_i, $i = 1, \dots, N_v$, and binary hidden units h_j, $j = 1, \dots, N_h$. Assuming binary visible units, the joint distribution takes the form:

$$P(\mathbf{v}, \mathbf{h}) = \exp\{-E(\mathbf{v}, \mathbf{h})\}/Z,$$
$$E(\mathbf{v}, \mathbf{h}) = -(\mathbf{h}^\top \mathbf{W}\mathbf{v} + \mathbf{c}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h}), \quad (1)$$

where $\mathbf{v} = [v_1, \dots, v_{N_v}]^\top$, $\mathbf{h} = [h_1, \dots, h_{N_h}]^\top$, and Z is the partition function which is a function of the model parameters $\mathbf{b}$, $\mathbf{c}$, $\mathbf{W}$. The posterior distributions for the hidden and visible units take the form

$$P(h_j = 1|\mathbf{v}) = \sigma\big(\sum_i w_{j,i} v_i + b_j\big), \quad (2)$$

$$P(v_i = 1|\mathbf{h}) = \sigma\big(\sum_j w_{j,i} h_j + c_i\big), \quad (3)$$

where $\sigma(x) = (1 + \exp\{-x\})^{-1}$. Using (2) and (3), inference is performed using block Gibbs sampling. The partial derivative of the data likelihood $P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h})$ with respect to each model parameter $\theta \in \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ takes the form (Hinton, 2002):

$$\frac{\partial}{\partial \theta} P(\mathbf{v}) = -(\mathbb{E}_{\mathbf{h}|\mathbf{v}} \frac{\partial}{\partial \theta} E(\mathbf{v}, \mathbf{h}) - \mathbb{E}_{\mathbf{v}', \mathbf{h}} \frac{\partial}{\partial \theta} E(\mathbf{v}', \mathbf{h})). \quad (4)$$

The first term in (4) can be evaluated analytically. However, the second term is intractable, but it can typically be approximated by running few iterations of block Gibbs sampling after initializing the visible units to the observation. This approximation is known as contrastive-divergence (CD) (Hinton, 2002).

2.2. Spike and Slab RBM

The ssRBM (Courville et al., 2011a;b) models a joint distribution over visible and hidden units, and a real valued slab variable s_j, $j = 1, \dots, N_h$, that is associated with each hidden unit. In this paper, we consider the version of Courville et al. (2011a). The energy function for the joint distribution of the ssRBM takes the following form:

$$E(\mathbf{v}, \mathbf{h}, \mathbf{s}) = -(\mathbf{s} \odot \mathbf{h})^\top \mathbf{W}\mathbf{v} + \frac{1}{2}\mathbf{v}^\top \big(\lambda \mathbf{I} + \mathrm{diag}(\Phi^\top \mathbf{h})\big)\mathbf{v} + \frac{\alpha}{2}\|\mathbf{s}\|_2^2 - \alpha\mu^\top(\mathbf{s} \odot \mathbf{h}) - \mathbf{b}^\top \mathbf{h} + \frac{\alpha}{2}\mu^{2\top}\mathbf{h} \quad (5)$$

where $\odot$ denotes element-wise product, $\mathrm{diag}(\cdot)$ denotes a diagonal matrix with the argument vector on its diagonal, $\mu^2 = \mu \odot \mu$, $\mathbf{I}$ is the identity matrix, $\mathbf{s} = [s_1, \dots, s_{N_h}]^\top$, and the model parameters are: $\lambda, \alpha \in \mathbb{R}$, $\mu \in \mathbb{R}^{N_h}$, $\Phi \in \mathbb{R}_+^{N_h \times N_v}$, $\mathbf{W} \in \mathbb{R}^{N_h \times N_v}$.

Inference in the ssRBM is performed using Gibbs sampling, from the following conditional distributions:

$$P(h_i = 1|\mathbf{v}) = \sigma(0.5\alpha^{-1}(\mathbf{W}_{i,\cdot}\mathbf{v})^2 + \mu_i \mathbf{W}_{i,\cdot}\mathbf{v} - 0.5\mathbf{v}^\top \mathrm{diag}(\Phi_{i,\cdot})\mathbf{v} + b_i), \quad (6)$$

$$P(s_i|\mathbf{v}, h_i) = \mathcal{N}((\alpha^{-1}\mathbf{W}_{i,\cdot}\mathbf{v} + \mu_i)h_i, \alpha^{-1}) \quad (7)$$

$$P(\mathbf{v}|\mathbf{s}, \mathbf{h}) = \mathcal{N}(C_{\mathbf{v}|\mathbf{s},\mathbf{h}}\mathbf{W}^\top(\mathbf{s} \odot \mathbf{h}), C_{\mathbf{v}|\mathbf{s},\mathbf{h}}) \quad (8)$$

where $\mathbf{W}_{i,\cdot}$ is the i^{th} row vector of $\mathbf{W}$ (the same is true for "$\Phi_{i,\cdot}$"), and $C_{\mathbf{v}|\mathbf{s},\mathbf{h}} = (\lambda \mathbf{I} + \mathrm{diag}(\Phi^\top \mathbf{h}))^{-1}$. Equations

(6)-(8) can be used iteratively using Gibbs sampling, and therefore learning can be performed using CD.

The conditional distribution of the observation given the hidden units takes the form:

$$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(C_{\mathbf{v}|\mathbf{h}}\mathbf{W}(\mu \odot \mathbf{h}), C_{\mathbf{v}|\mathbf{h}}), \tag{9}$$

where $C_{\mathbf{v}|\mathbf{h}} = (\lambda\mathbf{I} + \text{diag}(\Phi^\top\mathbf{h}) - \frac{1}{\alpha}\mathbf{W}^\top\text{diag}(\mathbf{h}))\mathbf{W})^{-1}$. This implies that the conditional covariance matrix $C_{\mathbf{v}|\mathbf{h}}$ is in general non-diagonal. This property of the ssRBM allows it to capture the covariance structure between the observations, which the GRBM does not model.

3. RTRBM revisited

The RTRBM describes the joint probability distribution of the visible units vector $\mathbf{v}_t \in \mathbb{R}^{N_v}$, and hidden units vector $\mathbf{h}_t \in \mathbb{R}^{N_h}$ at time step t, using a conditional RBM which depends on the hidden input $\mathbf{r}_{t-1}$. The RTRBM network is illustrated in Figure 1. The joint probability distribution of $\mathbf{v}_t, \mathbf{h}_t$ for any $t > 1$ (given $\mathbf{r}_{t-1}$) takes the following form:

$$P(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1}) = \frac{1}{Z_{r_{t-1}}} \exp\{-E(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1})\} \tag{10}$$

$$E(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1}) = -(\mathbf{h}_t^\top\mathbf{W}\mathbf{v}_t + \mathbf{c}^\top\mathbf{v}_t + \mathbf{b}^\top\mathbf{h}_t + \mathbf{h}_t^\top\mathbf{U}\mathbf{r}_{t-1}),$$

where $\mathbf{W} \in \mathbb{R}^{N_h \times N_v}$, $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{c} \in \mathbb{R}^{N_v}$, $\mathbf{b} \in \mathbb{R}^{N_h}$ are model parameters, and $Z_{\mathbf{r}_{t-1}}$ denotes a normalization factor which depends on $\mathbf{r}_{t-1}$ and the other model parameters (we used the subscript $\mathbf{r}_{t-1}$ since the dependency on the input $\mathbf{r}_{t-1}$ is the major difference compared to the RBM). For $t = 1$, the joint distribution of $\mathbf{v}_1, \mathbf{h}_1$ takes the form of a standard RBM with the hidden units biases $\mathbf{b}_{init} \in \mathbb{R}^{N_h}$.

The inputs $\mathbf{r}_t$ (where $t \in \{1, \ldots, T-1\}$) are obtained from a RNN given $\mathbf{v}_1, ..., \mathbf{v}_t$:

$$\mathbf{r}_t = \begin{cases} \sigma(\mathbf{W}\mathbf{v}_t + \mathbf{b} + \mathbf{U}\mathbf{r}_{t-1}), & \text{if } t > 1 \\ \sigma(\mathbf{W}\mathbf{v}_t + \mathbf{b}_{init}), & \text{if } t = 1 \end{cases} \tag{11}$$

where the logistic function $\sigma(x) = (1 + \exp(-x))^{-1}$ is applied to each element of its argument vector. The motivation for the choice of $\mathbf{r}_t$ is that using the RBM associated with time instant t, we have that $\mathbb{E}[\mathbf{h}_t|\mathbf{v}_t] = \mathbf{r}_t$; i.e., it is the expected value of the hidden units vector.

The joint probability distribution of the visible and hidden units of the RTRBM with length T takes the form:

$$P(\{\mathbf{v}_t, \mathbf{h}_t\}_{t=1}^T; \{\mathbf{r}_t\}_{t=1}^{T-1}) = \frac{\exp\{E(\{\mathbf{v}_t, \mathbf{h}_t\}_{t=1}^T; \{\mathbf{r}_t\}_{t=1}^{T-1})\}}{Z \cdot Z_{\mathbf{r}_1} \cdots Z_{\mathbf{r}_{T-1}}}$$

where Z denotes the normalization factor for the first RBM at $t = 1$, and where

$$E(\{\mathbf{v}_t, \mathbf{h}_t\}_{t=1}^T; \{\mathbf{r}_t\}_{t=1}^{T-1}) = -\big(\mathbf{h}_1^\top\mathbf{W}\mathbf{v}_1 + \mathbf{c}^\top\mathbf{v}_1 \tag{12}$$

$$+ \mathbf{b}_{init}^\top\mathbf{h}_1 + \sum_{t=2}^{T} \big(\mathbf{h}_t^\top\mathbf{W}\mathbf{v}_t + \mathbf{c}^\top\mathbf{v}_t + \mathbf{b}^\top\mathbf{h}_t + \mathbf{h}_t^\top\mathbf{U}\mathbf{r}_{t-1}\big)\big)$$

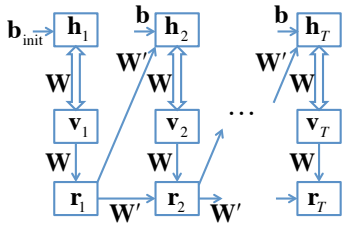


Figure 1. An illustration of the RTRBM.

3.1. Inference in the RTRBM

Given hidden inputs $\mathbf{r}_{t-1}$ ($t > 1$), the conditional distributions are factorized and takes the form:

$$P(h_{t,j} = 1|\mathbf{v}_t, \mathbf{r}_{t-1}) = \sigma(\sum_i w_{j,i}v_{t,i} + b_j + \sum_l u_{j,m}r_{t-1,m}),$$

$$P(v_{t,i} = 1|\mathbf{h}_t, \mathbf{r}_{t-1}) = \sigma(\sum_j w_{j,i}h_{t,j} + c_i), \tag{13}$$

For $t = 1$, the posterior of $\mathbf{h}_1$ given $\mathbf{v}_1$ has a similar form as (2), where $\mathbf{b}_{init}$ replaces $\mathbf{b}$. The above conditional probabilities can also be used to generate samples $\mathbf{v}_1, ..., \mathbf{v}_T$ (i.e., by repeatedly running Gibbs sampling for $t = 1, ..., T$).

Further, note that, given the hidden inputs $\mathbf{r}_1, ..., \mathbf{r}_{T-1}$, all the RBMs (corresponding to different time frames) are decoupled; thus, sampling can be performed using block Gibbs sampling for each RBM independently. This fact is useful in deriving the CD approximation for the RTRBM.

3.2. Learning in the RTRBM

In order to learn the parameters, we need to obtain the partial derivatives of the log-likelihood, $\log P(\mathbf{v}_1, ..., \mathbf{v}_T)$, with respect to the model parameters. Using the CD approximation to compute these derivatives requires the gradients of energy function (12) with respect to all the model parameters. We separate the energy function into the following two terms: $E = -\mathcal{H} - \mathcal{Q}_2$, where

$$\mathcal{H} = \mathbf{h}_1^\top\mathbf{W}\mathbf{v}_1 + \mathbf{c}^\top\mathbf{v}_1 + \mathbf{b}_{init}^\top\mathbf{h}_1 + \sum_{t=2}^{T} \mathbf{h}_t^\top\mathbf{W}\mathbf{v}_t + \mathbf{c}\mathbf{v}_t + \mathbf{b}^\top\mathbf{h}_t,$$

$$\mathcal{Q}_2 = \sum_{t=2}^{T} \mathbf{h}_t^T\mathbf{U}\mathbf{r}_{t-1}. \tag{14}$$

Taking the gradients of $\mathcal{H}$ with respect to the model parameters is straightforward, and therefore we focus on $\mathcal{Q}_2$. To compute the partial derivative of $\mathcal{Q}_2$ with respect to a model parameter θ, we first compute the gradient of $\mathcal{Q}_2$ with respect to $\mathbf{r}_t$, for $t = 1, \ldots, T-1$, which can be computed recursively using the backpropagation-through-time (BPTT) algorithm (Rumelhart et al., 1986). These gradients are then used with the chain rule to compute the derivatives with respect to all the model parameters. In the next subsection, we use the BPTT to derive the gradient with respect to $\mathbf{U}$. The other gradients can be computed similarly.

3.2.1. CALCULATING $\nabla_{\mathbf{r}_t}\mathcal{Q}_2$ USING BPTT

We observe that $\mathcal{Q}_2$ can be computed recursively using:

$$\mathcal{Q}_t = \sum_{\tau=t}^{T} \mathbf{h}_\tau^\top \mathbf{U}\mathbf{r}_{\tau-1} = \mathcal{Q}_{t+1} + \mathbf{h}_t^\top \mathbf{U}\mathbf{r}_{t-1} \tag{15}$$

where $\mathcal{Q}_{T+1} = 0$. Using the chain rule and (15), we have

$$\begin{aligned}\frac{\partial}{\partial r_{t,m}}\mathcal{Q}_{t+1} &= \sum_{m'} \frac{\partial \mathcal{Q}_{t+2}}{\partial r_{t+1,m'}}\frac{\partial r_{t+1,m'}}{\partial r_{t,m}} + \frac{\partial}{\partial r_{t,m}}(\mathbf{h}_{t+1}^\top \mathbf{U}\mathbf{r}_t) \\ &= \sum_{m'} \frac{\partial \mathcal{Q}_{t+2}}{\partial r_{t+1,m'}} r_{t+1,m'}(1-r_{t+1,m'})u_{m',m} \\ &+ \sum_{m'} h_{t+1,m'}u_{m',m} \end{aligned} \tag{16}$$

where $r_{t+1,m'}(1-r_{t+1,m'})$ is obtained from the partial derivative of the sigmoid function. Equation (16) can also be expressed in vector form using:

$$\nabla_{\mathbf{r}_t}\mathcal{Q}_{t+1} = \mathbf{U}^\top(\nabla_{\mathbf{r}_{t+1}}\mathcal{Q}_{t+2} \odot \mathbf{r}_{t+1} \odot (\mathbf{1}-\mathbf{r}_{t+1}) + \mathbf{h}_{t+1}), \tag{17}$$

where $\odot$ denotes element-wise product. Since $\mathcal{Q}_{t+1}$ is not a function of $\mathbf{r}_1, \ldots, \mathbf{r}_{t-1}$, we have that $\nabla_{\mathbf{r}_t}\mathcal{Q}_2 = \nabla_{\mathbf{r}_t}\mathcal{Q}_{t+1}$, and therefore the necessary partial derivatives can be computed recursively using (17).

3.2.2. CALCULATING THE PARTIAL DERIVATIVES WITH RESPECT TO THE MODEL PARAMETERS

In order to compute the derivatives with respect to $\mathbf{U}$, we use the chain rule and (15):

$$\begin{aligned}\frac{\partial \mathcal{Q}_2}{\partial u_{m,m'}} &= \sum_{t=2}^{T}\left(\frac{\partial \mathcal{Q}_{t+1}}{\partial r_{t,m}}\frac{\partial r_{t,m}}{\partial u_{m,m'}} + \frac{\partial}{\partial u_{m,m'}}(\mathbf{h}_t^\top \mathbf{U}\mathbf{r}_{t-1})\right) \\ &= \sum_{t=2}^{T}\left(\frac{\partial \mathcal{Q}_{t+1}}{\partial r_{t,m}} r_{t,m}(1-r_{t,m}) + h_{t,m}\right) r_{t-1,m'} \end{aligned} \tag{18}$$

where when taking the derivative of $\mathbf{h}_t^\top \mathbf{U}\mathbf{r}_{t-1}$ with respect to $u_{m,m'}$ we regard $\mathbf{r}_{t-1}$ as a constant, since the contribution of its derivative is factored in through $\nabla_{\mathbf{r}_{t-1}}\mathcal{Q}_t$.

Using the CD approximation with (16) and (18), it can be shown that (see supplemental material) the update rule for $\mathbf{U}$, that is related to $\mathcal{Q}_2$ (not including $\mathcal{H}$), takes the form:

$$\begin{aligned}\Delta_{\mathbf{U}}^{\mathcal{Q}_2} = \sum_{t=2}^{T} &\big(\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1}-\mathbf{r}_t) \\ &+ \mathbb{E}_{\mathbf{h}_t|\mathbf{v}_t,\mathbf{r}_{t-1}}[\mathbf{h}_t] - \mathbb{E}_{\mathbf{v}'_t,\mathbf{h}_t|\mathbf{r}_{t-1}}[\mathbf{h}_t]\big)\mathbf{r}_{t-1}^\top \end{aligned} \tag{19}$$

where

$$\begin{aligned}\mathcal{D}_t = &\mathbb{E}_{\mathbf{h}_t,\ldots,\mathbf{h}_T|\mathbf{v}_t,\ldots\mathbf{v}_T,\mathbf{r}_1\ldots,\mathbf{r}_{T-1}}[\nabla_{\mathbf{r}_{t-1}}\mathcal{Q}_t] \\ &- \mathbb{E}_{\mathbf{h}_t,\ldots,\mathbf{h}_T,\mathbf{v}'_t,\ldots\mathbf{v}'_T|\mathbf{r}_1\ldots,\mathbf{r}_{T-1}}[\nabla_{\mathbf{r}_{t-1}}\mathcal{Q}_t]. \end{aligned} \tag{20}$$

Intuitively speaking, the first term is the expectation over "data distribution", and the second term is the expectation over "model distribution" (conditioned on $\{\mathbf{r}_t\}_{t=1}^{T-1}$). Similarly to (17), $\mathcal{D}_t$ can be computed recursively using:

$$\begin{aligned}\mathcal{D}_{t+1} = &\mathbf{U}^\top(\mathcal{D}_{t+2} \odot \mathbf{r}_{t+1} \odot (\mathbf{1}-\mathbf{r}_{t+1}) \\ &+ \mathbb{E}_{\mathbf{h}_{t+1}|\mathbf{v}_{t+1},\mathbf{r}_t}[\mathbf{h}_{t+1}] - \mathbb{E}_{\mathbf{v}'_{t+1},\mathbf{h}_{t+1}|\mathbf{r}_t}[\mathbf{h}_{t+1}]), \end{aligned} \tag{21}$$

and $\mathcal{D}_{T+1} = 0$ (see supplementary material for details).

The model parameters $\theta \in \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{b}_{init}, \mathbf{c}\}$ is updated via gradient ascent (e.g., $\theta := \theta + \eta\Delta_\theta^{\mathcal{H}+\mathcal{Q}_2}$) where

$$\begin{aligned}\Delta_\theta^{\mathcal{H}+\mathcal{Q}_2} = &\mathbb{E}_{\{\mathbf{h}_t\}_{t=1}^T|\{\mathbf{v}_t,\mathbf{r}_t\}_{t=1}^T}[\nabla_\theta \mathcal{H}] \\ &- \mathbb{E}_{\{\mathbf{h}_t,\mathbf{v}'_t\}_{t=1}^T|\{\mathbf{r}_t\}_{t=1}^T}[\nabla_\theta \mathcal{H}] + \Delta_\theta^{\mathcal{Q}_2} \end{aligned} \tag{22}$$

For the other model parameters, we have $\Delta_{\mathbf{c}}^{\mathcal{Q}_2} = 0$, and

$$\Delta_{\mathbf{W}}^{\mathcal{Q}_2} = \sum_{t=1}^{T-1}(\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1}-\mathbf{r}_t))\mathbf{v}_t^\top \tag{23}$$

$$\Delta_{\mathbf{b}}^{\mathcal{Q}_2} = \sum_{t=2}^{T-1}(\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1}-\mathbf{r}_t)) \tag{24}$$

$$\Delta_{\mathbf{b}_{init}}^{\mathcal{Q}_2} = \mathcal{D}_2 \odot \mathbf{r}_1 \odot (\mathbf{1}-\mathbf{r}_1) \tag{25}$$

4. Structured RTRBM

The energy function at each stage in the RTRBM as defined in Eq. (10) assumes full connectivity between all the pairs of hidden units and inputs from previous time-steps, and pairs of visible and hidden units. Our proposed SRTRBM uses a dependency graph to define the set of allowed interactions. We start by developing the SRTRBM assuming that the graph structure is known, and then relax the energy function in order to learn the topology from the data.

4.1. SRTRBM with a known dependency graph

In order to develop the SRTRBM, we use a graph structure to define block masking matrices that are associated with the model parameters $\mathbf{M}$ and $\mathbf{U}$, where the sparsity pattern of the making matrices is specified by the graph. Intuitively speaking, we partition visible units, and each subset of the partition is associated with a *node*, as well as its corresponding hidden units.

Specifically, we assume an undirected graph structure $G = (V, E)$, where $V = \{1, \ldots, |V|\}$ denotes the set of nodes, and E denotes the set of undirected edges. We associate with each node $\epsilon \in V$ a subset of visible units indices $C_\epsilon^v \subset \{1, \ldots, N_v\}$, where N_v denotes the number of visible units (we note that a group can be comprised of a single observation as well), such that $\cup_\epsilon C_\epsilon^v = \{1, \ldots, N_v\}$, and $\cap_\epsilon C_\epsilon^v = \emptyset$ (i.e. the subsets are disjoint and each visible unit is assigned to a node). Similarly, we define a set of hidden units indices $C_\epsilon^h \subset \{1, \ldots, N_h\}$, where N_h denotes the number of hidden units, such that $\cup_\epsilon C_\epsilon^h = \{1, \ldots, N_h\}$, and $\cap_\epsilon C_\epsilon^h = \emptyset$. We rearrange the order of the vector $\mathbf{v}_t$ such that $\mathbf{v} = [\mathbf{v}^{1\top}, \ldots, \mathbf{v}^{\epsilon\top}, \ldots, \mathbf{v}^{|V|\top}]^\top$, where

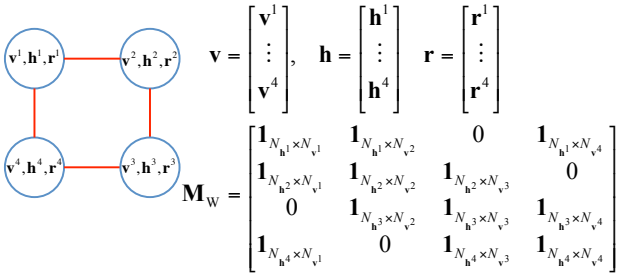


Figure 2. Construction of the masking matrix $\mathbf{M_W}$, given the graph structure (N_x denotes the dimension of the vector x)

$\mathbf{v}^\epsilon$ is a vector containing all the visible units assigned to node ϵ, and similarly we rearrange the order of the vectors $\mathbf{h}_t$, $\mathbf{r}_t$, such that $\mathbf{h} = [\mathbf{h}^{1\top}, \ldots, \mathbf{h}^{\epsilon\top}, \ldots, \mathbf{h}^{|V|\top}]^\top$, $\mathbf{r} = [\mathbf{r}^{1\top}, \ldots, \mathbf{r}^{\epsilon\top}, \ldots, \mathbf{r}^{|V|\top}]^\top$, where $\mathbf{h}^\epsilon$, $\mathbf{r}^\epsilon$ are the vectors containing all the hidden units and hidden inputs from the previous time-step assigned to node ϵ respectively. We define the energy function of SRTRBM for each $t > 1$ as:

$$E(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1}) = -(\mathbf{h}_t^\top (\mathbf{W} \odot \mathbf{M_W})\mathbf{v}_t + \mathbf{c}^\top \mathbf{v}_t + \mathbf{b}^\top \mathbf{h}_t + \mathbf{h}_t^\top (\mathbf{U} \odot \mathbf{M_U})\mathbf{r}_{t-1}), \quad (26)$$

where $\mathbf{r}_t = \sigma((\mathbf{W} \odot \mathbf{M_W})\mathbf{v}_t + \mathbf{b} + (\mathbf{U} \odot \mathbf{M_U})\mathbf{r}_{t-1})$ for $t > 1$, and $\mathbf{r}_1 = \sigma((\mathbf{W} \odot \mathbf{M_W})\mathbf{v}_1 + \mathbf{b}_{init})$. For $t = 1$,

$$E(\mathbf{v}_1, \mathbf{h}_1) = -(\mathbf{h}_1^\top (\mathbf{W} \odot \mathbf{M_W})\mathbf{v}_1 + \mathbf{c}^\top \mathbf{v}_1 + \mathbf{b}_{init}^\top \mathbf{h}_1),$$

where $\mathbf{M_W} \in \mathbb{R}^{N_h \times N_v}$ is a $|V| \times |V|$ block masking matrix such that the ji^{th} block has the dimensions $|C_j^h| \times |C_i^v|$, and the ji^{th} block is set to all ones if there is an edge connecting nodes i and j, and otherwise to all zeros. All the "diagonal" blocks (i.e., the ii^{th} blocks) are set to all ones. Similarly, $\mathbf{M_U} \in \mathbb{R}^{N_h \times N_h}$ is a $|V| \times |V|$ block masking matrix, where the ji^{th} block has the dimensions $|C_j^h| \times |C_i^h|$, and the ji^{th} block is set to all ones if there is an edge connecting nodes i and j, and otherwise to all zeros. All the diagonal blocks are set to all ones as well. The construction of the masking matrix $\mathbf{M_W}$ is illustrated in Figure 2.

The assignment of observations to groups can be performed in several ways. If some prior knowledge is available (e.g., if some observations belong to the same node in a sensor network), it can be used to determine the groups. Alternatively, one can assign a single observation to each node.

Inference and learning are performed similarly to the RTRBM, where $\mathbf{W}$ and $\mathbf{U}$ are replaced by $\mathbf{W} \odot \mathbf{M_W}$ and $\mathbf{U} \odot \mathbf{M_U}$ respectively. When updating the parameter matrices $\mathbf{W}$ and $\mathbf{U}$, we simply multiply the partial derivatives by their corresponding masking matrix.

4.2. Learning the dependency graph from the data

In order to learn the graph structure from the data, we replace the blocks of the masking matrices $\mathbf{M_W}$ and $\mathbf{M_U}$ with logistic functions $\sigma_\delta(\nu) = (1 + \exp\{-\delta\nu\})^{-1}$, where the ji^{th} block of both masking matrices is set to $\sigma_\delta(\nu_{ji})$, and where $\delta \geq 1$ is a hyper-parameter. Learning the graph structure now corresponds to learning the logistic parameters ν_{ji}. Setting the hyper-parameter δ to values larger than 1 tends to speed up the convergence of the learning algorithm, since the logistic function saturates for smaller magnitudes of ν. We used $\delta = 8$ throughout this work.

We use CD to learn the logistic parameters. Using (26) we have that the partial derivative of the SRTRBM energy function with respect to ν_{ji} takes the form:

$$\frac{\partial}{\partial \nu_{ji}} E(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1}) = -\sum_t \big[\mathbf{h}_t^{j\top} \mathbf{W}^{ji} \mathbf{v}_t^i + \mathbf{h}_t^{j\top} \mathbf{U}^{ji} \mathbf{r}_{t-1}^i + (\nabla_{\mathbf{r}_t} \mathcal{Q}_{t+1}^j \odot \mathbf{r}_t^j \odot (\mathbf{1} - \mathbf{r}_t^j))^\top (\mathbf{W}^{ji} \mathbf{v}_t^i + \mathbf{U}^{ji} \mathbf{r}_{t-1}^i)\big] \times \delta \sigma_\delta(\nu_{ji})(\mathbf{1} - \sigma_\delta(\nu_{ji})) \quad (27)$$

where $\mathbf{W}^{ji}$ and $\mathbf{U}^{ji}$ denote the ji^{th} block of $\mathbf{W}$ and $\mathbf{U}$ respectively.

Sparsity regularization. In order to learn sparsely connected graph structures, we propose to regularize the logistic parameters ν_{ji} such they are encouraged to be close to some negative hyper-parameter. Specifically, we add the penalty term $-\beta \sum_{j \neq i} |\nu_{ji} - \kappa|$, where κ is a negative constant (we used $\kappa = -3$ throughout this work), and β is the regularization parameter. This method can also be interpreted as using a Laplace distribution prior with negative mean for ν_{ji}. Note that this regularization is different from the conventional sparsity regularization imposed on the activation of hidden units (Lee et al., 2008).

5. Spike and slab RTRBM and SRTRBM

Previous applications of the RTRBM to modeling of time-series with real valued observations used a GRBM for each stage in the network. Here we propose to use a ssRBM instead of the GRBM, since it is known to provide better modeling for the conditional covariance. The hidden input from the previous time-step is also replaced with the expected value of the hidden units at each stage in the network. We do not add any additional terms related to the slab variables. Following (5) and (10), we have that the probability distribution for each time-step takes the form:

$$P(\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t; \mathbf{r}_{t-1}) = \frac{1}{Z_{r_{t-1}}} \exp\{-E(\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t; \mathbf{r}_{t-1})\}$$

$$E(\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t; \mathbf{r}_{t-1}) = -(\mathbf{s}_t \odot \mathbf{h}_t)^\top \mathbf{W}\mathbf{v}_t + \frac{1}{2}\mathbf{v}_t^\top \big(\lambda \mathbf{I} + \mathrm{diag}(\Phi^\top \mathbf{h}_t)\big)\mathbf{v}_t + \frac{\alpha}{2}\|\mathbf{s}_t\|_2^2 - \alpha\mu^\top (\mathbf{s}_t \odot \mathbf{h}_t) - \mathbf{b}^\top \mathbf{h}_t + \frac{\alpha}{2}\mu^{2\top}\mathbf{h}_t - \mathbf{h}_t^\top \mathbf{U}\mathbf{r}_{t-1}, \quad (28)$$

where $r_{t,i} = \sigma(0.5\alpha^{-1}(\mathbf{W}_{i,\cdot}\mathbf{v}_t)^2 + \mu_i \mathbf{W}_{i,\cdot}\mathbf{v}_t - 0.5\mathbf{v}_t^\top \mathrm{diag}(\Phi_{i,\cdot})\mathbf{v}_t + b_i + \mathbf{U}_{i,\cdot}\mathbf{r}_{t-1})$ for $t > 1$, and $r_{1,i} = \sigma(0.5\alpha^{-1}(\mathbf{W}_{i,\cdot}\mathbf{v}_1)^2 + \mu_i \mathbf{W}_{i,\cdot}\mathbf{v}_1 - 0.5\mathbf{v}_1^\top \mathrm{diag}(\Phi_{i,\cdot})\mathbf{v}_1 + b_{init,i})$. The model parameters are $\{\mathbf{W}, \mathbf{U}, \mathbf{b}, \Phi, \mu, \lambda, \alpha\}$.

The joint probability takes the form:

$$P(\{\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t\}_{t=1}^T; \{\mathbf{r}_t\}_{t=1}^{T-1}) \propto \exp\{E(\{\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t\}_{t=1}^T)\},$$

where $E(\{\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t\}_{t=1}^T) = -\mathcal{H} - \mathcal{Q}_2$, with

$$\begin{aligned}\mathcal{H} = (\mathbf{b}_{init} - \mathbf{b})^\top \mathbf{h}_1 + \sum_{t=1}^{T} \big[&(\mathbf{s}_t \odot \mathbf{h}_t)^\top \mathbf{W} \mathbf{v}_t \\ &- \frac{1}{2}\mathbf{v}_t^\top(\lambda \mathbf{I} + \text{diag}(\Phi^\top \mathbf{h}_t))\mathbf{v}_t - \frac{\alpha}{2}\|\mathbf{s}_t\|_2^2 \\ &+ \alpha \mu^\top (\mathbf{s}_t \odot \mathbf{h}_t) + \mathbf{b}^\top \mathbf{h}_t - \frac{\alpha}{2}\mu^{2\top}\mathbf{h}_t\big], \end{aligned} \tag{29}$$

and $\mathcal{Q}_2$ defined as in (14).

Inference can be performed using Gibbs sampling, where $\mathbf{v}_t, \mathbf{h}_t, \mathbf{s}_t$ are sampled similarly to (6)-(8), and learning can be performed using CD. The gradients with respect to each of the model parameters related to $\mathcal{H}$ are straightforward to compute, whereas the gradients related to $\mathcal{Q}_2$ are computed using the same approach used for the RTRBM. First, the gradients with respect to $\mathbf{r}_t$ are computed using (17), and subsequently the gradients with respect to each of the model parameters are computed using the chain rule.

5.1. Spike-and-slab SRTRBM

In order to learn structure in real valued time-series using the spike-and-slab SRTRBM, we define the masking matrices $\mathbf{M_W}$ and $\mathbf{M_U}$ similarly to the previous section. We then replace the model matrices $\mathbf{W}$, $\mathbf{U}$, and Φ, with $\mathbf{W} \odot \mathbf{M_W}$, $\mathbf{U} \odot \mathbf{M_U}$, and $\Phi \odot \mathbf{M_W}$ respectively. Learning of the logistic parameters ν_{ji} is performed using CD (similarly as in Section 4.2). Using (28) we have that the partial derivative of the spike-and-slab SRTRBM energy function with respect to ν_{ji} takes the form:

$$\begin{aligned}\frac{\partial}{\partial \nu_{ji}} E(\mathbf{v}_t, \mathbf{h}_t; \mathbf{r}_{t-1}) = -\sum_t \big[&(\mathbf{h}_t^j \odot \mathbf{s}_t^j)^\top \mathbf{W}^{ji} \mathbf{v}_t^i \\ &+ \mathbf{h}_t^{j\top}(-0.5\Phi^{ji}(\mathbf{v}_t^i \odot \mathbf{v}_t^i) + \mathbf{U}^{ji}\mathbf{r}_{t-1}^i) \\ &+ (\nabla_{\mathbf{r}_t}\mathcal{Q}_{t+1}^j \odot \mathbf{r}_t^j \odot (\mathbf{1} - \mathbf{r}_t^j) \odot (\mu^j + \alpha^{-1}(\widetilde{\mathbf{W}}\mathbf{v}_t)^j))^\top \mathbf{W}^{ji}\mathbf{v}_t^i \\ &+ (\nabla_{\mathbf{r}_t}\mathcal{Q}_{t+1}^j \odot \mathbf{r}_t^j \odot (\mathbf{1} - \mathbf{r}_t^j))^\top (\mathbf{U}^{ji}\mathbf{r}_{t-1}^i - 0.5\Phi^{ji}(\mathbf{v}_t^i \odot \mathbf{v}_t^i))\big] \\ &\times \delta\sigma_\delta(\nu_{ji})(\mathbf{1} - \sigma_\delta(\nu_{ji})) \end{aligned} \tag{30}$$

where $\widetilde{\mathbf{W}} = \mathbf{W} \odot \mathbf{M_W}$. The learning scheme for the spike-and-slab SRTRBM is summarized in Algorithm 1.

Algorithm 1 Contrastive Divergence training iteration for the spike-and-slab SRTRBM.

Input: training sequence $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and number of CD iterations N_{CD}.

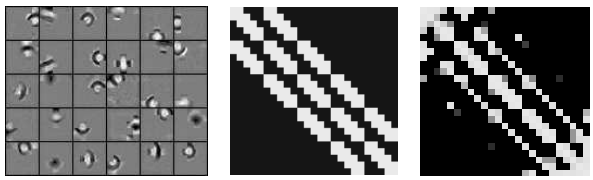
1. Set $\widetilde{\mathbf{W}} = \mathbf{W} \odot \mathbf{M_W}$, $\widetilde{\mathbf{U}} = \mathbf{U} \odot \mathbf{M_U}$, $\widetilde{\Phi} = \Phi \odot \mathbf{M_W}$.
2. For $t = 1, \ldots, T$
   - For each $i = 1, \ldots, N_h$, compute: $\mathbf{r}_{t,i} = \sigma(0.5\alpha^{-1}(\widetilde{\mathbf{W}}_{i,\cdot}\mathbf{v}_t)^2 + \mu_i \widetilde{\mathbf{W}}_{i,\cdot}\mathbf{v}_t - 0.5\mathbf{v}_t^\top \text{diag}(\widetilde{\Phi}_{i,\cdot})\mathbf{v}_t + b'_{t,i}))$, where $b'_{1,i} = b_{init,i}$ and $b'_{t,i} = b_i + \widetilde{\mathbf{U}}_{i,\cdot}\mathbf{r}_{t-1}$ for $t > 1$.
   - Run N_{CD} Gibbs sampling iterations to obtain $\mathbf{v}_t^{(n)}, \mathbf{h}_t^{(n)}, \mathbf{s}_t^{(n)}, n = 0, \ldots, N_{CD}$.
3. Approximate $\mathcal{D}_t$ recursively using $\mathbf{h}_t^{(0)}$, $\mathbf{h}_t^{(N_{CD})}$, $t = T, \ldots, 2$ using (21), using $\widetilde{\mathbf{U}}$ instead of $\mathbf{U}$.
4. Compute approximate gradients related to $\mathcal{Q}_2$ with respect to the model parameters:
   - $\Delta_{\mathbf{W}}^{\mathcal{Q}_2} = \sum_{t=1}^{T-1}((\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1} - \mathbf{r}_t)) \odot (\alpha^{-1}\widetilde{\mathbf{W}}\mathbf{v}_t + \mu))\mathbf{v}_t^\top$
   - $\Delta_{\Phi}^{\mathcal{Q}_2} = \sum_{t=1}^{T-1} -0.5(\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1} - \mathbf{r}_t))\mathbf{v}_t^{2\top}$
   - $\Delta_{\mu}^{\mathcal{Q}_2} = \sum_{t=1}^{T-1}(\mathcal{D}_{t+1} \odot \mathbf{r}_t \odot (\mathbf{1} - \mathbf{r}_t)) \odot (\widetilde{\mathbf{W}}\mathbf{v}_t)$
   - Compute $\Delta_{\mathbf{U}}^{\mathcal{Q}_2}$, $\Delta_{\mathbf{b}}^{\mathcal{Q}_2}$, and $\Delta_{\mathbf{b}_{init}}^{\mathcal{Q}_2}$ using (19), (24), and (25) respectively.
5. For each $\theta \in \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{b}_{init}, \Phi, \mu, \lambda\}$ update using
$$\Delta_\theta^{\mathcal{H}+\mathcal{Q}_2} = \big(\frac{\partial}{\partial\theta}\mathcal{H}(\{\mathbf{v}_t^{(0)}, \mathbf{h}_t^{(0)}\}_{t=1}^T) - \frac{\partial}{\partial\theta}\mathcal{H}(\{\mathbf{v}_t^{(N_{CD})}, \mathbf{h}_t^{(N_{CD})}\}_{t=1}^T) + \Delta_\theta^{\mathcal{Q}_2}\big) \odot \mathbf{M}_\theta$$
   where $\mathcal{H}$ is given in (29), $\mathbf{M}_\Phi = \mathbf{M_W}$, $\mathbf{M_b} = \mathbf{M}_{\mathbf{b}_{init}} = \mathbf{M}_\mu = \mathbf{M}_\lambda = \mathbf{1}$, and $\Delta_\lambda^{\mathcal{Q}_2} = 0$.
6. Set any negative element of Φ to zero.
7. Update ν_{ji} using (30) and the sparsity regularization.

6. Experimental results

6.1. Simulations using synthetic data

Here we use synthetic videos of 3 bouncing balls, where the pixels are binary valued. We generated 4000 videos of size 30×30 pixels and duration of 100 time-steps for training, using the code that is available online and was described in Sutskever et al. (2008). We generated another 200 such videos for testing. We defined the SRTRBM groups by partitioning each frame into a 5×5 grid, and assigning the pixels within each block to a unique node in the graph. We used the same total number of hidden units for the RTRBM and SRTRBM. For the SRTRBM, the number of hidden units assigned to each SRTRBM groups was identical, and the regularization hyper-parameter to $\beta = 0.001$. The number of CD iterations during training was set to 25.

In Figure 3(a), we show 30 of the bases that were learned using the SRTRBM. Comparing to the bases that were presented in previous works (Sutskever et al., 2008; Boulanger-Lewandowski et al., 2012), the bases shown in Figure 3(a) are more spatially localized. In Table 1, we



(a) Learned bases (b) Neighbors (c) Learned mask

Figure 3. (a) The bases learned using SRTRBM for the videos of bouncing balls, (b) the mask $\mathbf{M_U}$ corresponding to a dependency graph in which only neighboring nodes (i.e., adjacent regions in the grid) are connected, (c) the learned mask $\mathbf{M_U}$.

N_h	2500	3750
RTRBM	4.00 ± 0.35	3.88 ± 0.33
SRTRBM	$\mathbf{3.58 \pm 0.35}$	$\mathbf{3.31 \pm 0.33}$

Table 1. Average prediction error for the bouncing balls dataset. Intervals represent the standard error of the mean.

compare the average squared prediction error per frame over the 200 test videos, for the RTRBM and SRTRBM with different number of hidden units. We used 5 iterations per frame to hallucinate the prediction. It can be seen that the SRTRBM outperforms the RTRBM, and that using the SRTRBM instead of the RTRBM becomes more important when the number of hidden units increases. In an additional control experiment, we also tried to use ℓ_1 regularization for the model matrix $\mathbf{W}$ of the RTRBM; however, it did not improve the performance of the RTRBM.

In Figure 3(b) we show the sparse adjacency matrix for a graph in which only the neighboring nodes (adjacent regions in the 5x5 grid) are connected. In Figure 3(c), we show the learned masking matrix $\mathbf{M_U}$ (i.e., brighter intensities correspond to larger values, and dark intensity values correspond to smaller values). The results suggest that the learned graph captures the dependency between adjacent blocks, as is expected for the bouncing balls dataset.

6.2. Simulations using motion capture data

In this experiment, we used the CMU motion capture dataset[1], which contains the joint angles for different motion types. We followed Taylor et al. (2011) and used the 33 running and walking sequences of subject 35 (23 walking sequences, and 10 running sequences). We partitioned the 33 sequences into two groups: the first had 31 sequences, and the second had 2 sequences (one walking, and the other running). We used the following preprocessing on the data. First, we removed all the joint angles which were constant throughout all the time-series (after which we were left with 58 angles). Subsequently, we subsampled each sequence by a factor of 4, and normalized the angles by first subtracting the mean from each joint and dividing by the standard deviation. We evaluated the performance in this normalized space.

[1] http://mocap.cs.cmu.edu/

	walking	running
RNN	21.67	19.05
Gaussian-RTRBM	14.41 ± 0.38	10.91 ± 0.27
ss-RTRBM	8.45 ± 0.05	6.22 ± 0.04
ss-SRTRBM (no grouping)	8.32 ± 0.08	$\mathbf{5.84 \pm 0.05}$
ss-SRTRBM (groups)	$\mathbf{8.13 \pm 0.06}$	$\mathbf{5.88 \pm 0.05}$

Table 2. Average prediction error obtained for the motion capture dataset using different methods ("no groupings": one observation per node; "groups": all the observations associated with each body part are assigned to the same node).

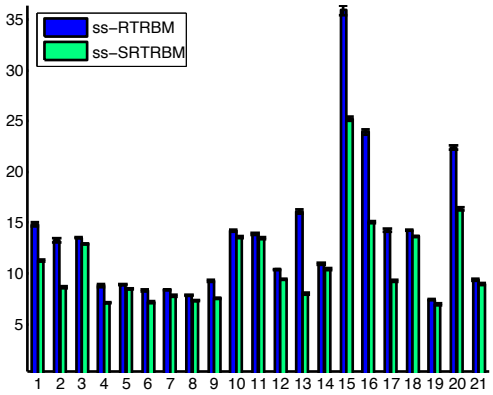


Figure 4. The prediction error for each motion capture testing sequence, obtained using the spike-and-slab RTRBM (ss-RTRBM) and the spike-and-slab SRTRBM (ss-SRTRBM)

We used two different approaches to assign the observations to SRTRBM groups. First, we assigned each observation to a group (i.e., no grouping), and the second approach was to assign all the angles associated with one of 27 body parts (e.g. left-foot, right-hand, etc.) to a single group. The number of hidden units that were assigned to each node in the SRTRBM was 10 times the number of observations assigned to that node, and we used the same total number of hidden units for the RTRBM.

We used a fixed learning rate of 10^{-3} with 10 CD iterations for training the Gaussian RTRBM, and a single CD iteration for training the spike-and-slab-based methods since we observed that when using more iterations the spike-and-slab-based algorithms often diverged. We hypothesize that this is because the probability distribution of the ssRBM is not guaranteed to integrate to 1. The spike-and-slab hyperparameter α was set to 1. The SRTRBM regularization parameter β was set to 5×10^{-3} when assigning each observation to an independent node, and 10^{-2} when assigning observations to groups based on the body parts. We used a single Gibbs sampling iteration per time-step to predict the normalized joint angles, since it produced the best results for all the compared algorithms. We averaged the prediction error over 200 trials.

In Table 2, we report the average prediction error obtained

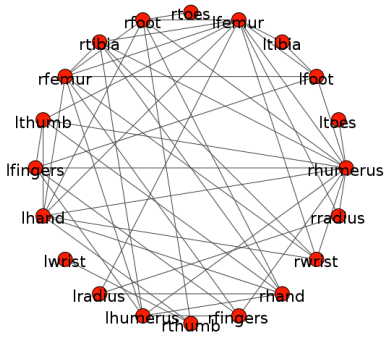


Figure 5. The sub-graph related to the extremities and major bones in the hands and legs, learned using the spike-and-slab SRTRBM for the motion capture dataset.

when training with the time-series from the first group, and testing with the time-series from the second, using RNN, Gaussian RTRBM, spike-and-slab RTRBM, and spike-and-slab SRTRBM with the two different groupings described before. It can be seen that the spike-and-slab-based methods improve over the Gaussian RTRBM significantly, and that the SRTRBM improves over the RTRBM. For the spike-and-slab SRTRBM, "no grouping" achieved comparable (or only slightly worse) performance to grouping-based method (based on prior knowledge).

In Figure 5, we show the parts of the dependency graph which are related to the extremities and major bones in the legs and hands. We used the sequences from the first group to learn the graph using the SRTRBM, where we assigned all the angles associated with each body part to the same node. It can be observed that many of the edges satisfy adjacency relationships in the human body (e.g. left-thumb and left-hand, left-thumb and left-fingers, left-foot and left-tibia, left-femur and left-tibia, right-tibia and right-femur, right-foot and right-femur, right-foot and right-toes, etc.).

6.3. Weather modeling

In this experiment, we used a dataset of historical weather records available at the National Climatic Data Center[2]. We used the maximum and minimum daily temperature measurements from 120 weather stations, and we normalized the data using the same procedure described in the previous section. We assigned the minimum and maximum temperatures measured at each station to the same node in the SRTRBM, and assigned 10 hidden units per observation. We used a learning rate of 5×10^{-5}, and SRTRBM regularization parameter $\beta = 10^{-2}$. We used the spike-and-slab parameter $\alpha = 8.5$ and $\alpha = 1$ for the non-structured and structured versions of the RTRBM respectively. We extracted 6 training sequences of length 50 time-steps each for training, and 5 such sequences for testing. The prediction error was computed by running a single Gibbs sampling iteration for each time-step, and averaging over 200 runs. The prediction error results are shown in Table 3, where it can be seen that the spike-and-slab-based methods significantly outperform the Gaussian RTRBM, and that the SRTRBM outperforms the RTRBM.

[2]ftp://ftp.ncdc.noaa.gov/pub/data/ushcn/daily

	GRTRBM	ss-RTRBM	ss-SRTRBM
1	33.69 ± 0.77	26.24 ± 0.12	$\mathbf{23.49 \pm 0.16}$
2	42.75 ± 0.97	35.16 ± 0.14	$\mathbf{32.91 \pm 0.17}$
3	38.68 ± 1.00	$\mathbf{30.81 \pm 0.14}$	$\mathbf{30.87 \pm 0.22}$
4	34.46 ± 0.86	25.55 ± 0.1	$\mathbf{24.55 \pm 0.16}$
5	35.59 ± 0.76	27.62 ± 0.11	$\mathbf{24.51 \pm 0.14}$

Table 3. Average prediction error for the historical weather records dataset, for each of the 5 testing sequences.

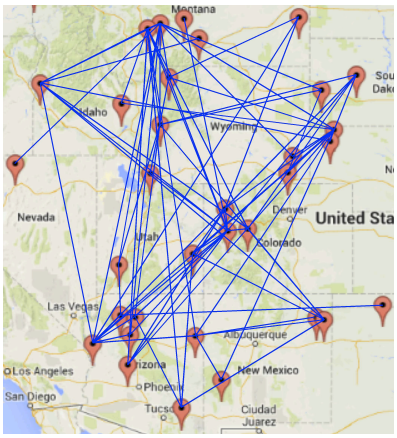


Figure 6. The graph corresponding to a subset of the 120 weather stations, learned using the historical weather records dataset.

In Figure 6, we present a subset of the learned graph for 24 stations out of the 120 that are located at the western part of the US. We used both the training and testing sequences to train the model. It can be observed that the links capture the neighborhood relationship as well as some longer range relationships.

7. Conclusions

We developed a new approach, the structured recurrent temporal restricted Boltzmann machine, to learn the structure of time-series signals. In addition, we proposed the spike-and-slab RTRBM, and combined it with the SRTRBM to learn the structure of real-valued time-series datasets. Our experimental results using several synthetic and real datasets verify that the SRTRBM is able to recover the structure of the datasets and outperform the RTRBM, particularly when the size of the training set is small. Furthermore, the spike-and-slab version of the RTRBM significantly outperforms the Gaussian RTRBM. We hope that our approach will inspire more research on deep structural learning models for temporal modeling.

Acknowledgments

This work was supported in part by ONR N000141310762, NSF CPS-0931474, and Google Faculty Research Award.

References

Banerjee, O., Ghaoui, L. E., and D'aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. Advances in optimizing recurrent networks. In *ICASSP*, 2013.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *ICML*, 2012.

Courville, A. C., Bergstra, J., and Bengio, Y. Unsupervised models of images by spike and-slab RBMs. In *ICML*, 2011a.

Courville, A. C., Bergstra, J., and Bengio, Y. A spike and slab restricted Boltzmann machine. In *AISTATS*, 2011b.

Dahl., G. E., Ranzato, M., Rahman, M. A., and Hinton, G. E. Phone recognition with the mean-covariance restricted Boltzmann machine. In *NIPS*, 2010.

Doshi, F., Wingate, D., Tenenbaum, J. B., and Roy, N. Infinite dynamic Bayesian networks. In *ICML*, 2011.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, (5):432441, 2008.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771–1800, 2002.

Hinton, G. E. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science*, 313: 504–507, 2006.

Lauritzen, S. L. *Graphical Models*. Oxford University Press, 1996.

Lee, H., Ekanadham, C., and Ng, A. Y. Sparse deep belief net model for visual area V2. In *NIPS*. 2008.

Martens, J. and Sutskever, I. Learning recurrent neural networks with Hessian-free optimization. In *ICML*, 2011.

Mohan, K., Chung, M., Han, S., Witten, D., Lee, S. I., and M, F. Structured learning of Gaussian graphical models. In *NIPS*. 2012.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *ICML*, volume 28, 2013.

Prabhakar, K., Oh, S. M., Wang, P., Abowd., G. D., and Rehg, J. M. Temporal causality for the analysis of visual events. In *CVPR*, 2010.

Rabiner, L. and Juang, B.-H. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

Ranzato, M. and Hinton, G. E. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *CVPR*, 2010.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. In Rumelhart, D. E. and Mcclelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pp. 318–362. MIT Press, Cambridge, MA, 1986.

Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D. E., McClelland, J. L., et al. (eds.), *Parallel Distributed Processing: Volume 1: Foundations*, pp. 194–281. MIT Press, Cambridge, 1987.

Songsiri, J. and Vandenberghe, L. Topology selection in graphical models of autoregressive processes. *Journal of Machine Learning Research*, 11:2671–2705, 2010.

Sutskever, I. and Hinton, G. E. Learning multilevel distributed representations for high-dimensional sequences. In *AISTATS*, 2007.

Sutskever, I., Hinton, G. E., and Graham, T. W. The recurrent temporal restricted Boltzmann machine. In *NIPS*, 2008.

Sutskever, I., Martens, J., Dahl, G. E., and Hinton, G. E. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

Taylor, G. W., Hinton, G. E., and Roweis, S. T. Two distributed-state models for generating high-dimensional time series. *Journal of Machine Learning Research*, 12: 1025–1068, 2011.