Echo-State Conditional Restricted Boltzmann Machines

Sotirios P. Chatzis
Department of Electrical Engineering, Computer Engineering, and Informatics
Cyprus University of Technology
Limassol 3603, Cyprus
soteri0s@mac.com

Abstract

Restricted Boltzmann machines (RBMs) are a powerful generative modeling technique, based on a complex graphical model of hidden (latent) variables. Conditional RBMs (CRBMs) are an extension of RBMs tailored to modeling temporal data. A drawback of CRBMs is their consideration of linear temporal dependencies, which limits their capability to capture complex temporal structure. They also require many variables to model long temporal dependencies, a fact that might provoke overfitting proneness. To resolve these issues, in this paper we propose the echo-state CRBM (ES-CRBM): our model uses an echo-state network reservoir in the context of CRBMs to efficiently capture long and complex temporal dynamics, with much fewer *trainable* parameters compared to conventional CRBMs. In addition, we introduce an (implicit) mixture of ES-CRBM experts (im-ES-CRBM) to enhance even further the capabilities of our ES-CRBM model. The introduced im-ES-CRBM allows for better modeling temporal observations which might comprise a number of latent or observable subpatterns that alternate in a dynamic fashion. It also allows for performing sequence segmentation using our framework. We apply our methods to sequential data modeling and classification experiments using public datasets.

Introduction

Restricted Boltzmann machines (RBMs) (Smolensky 1986) are a popular class of two-layer undirected graphical models that model observations by means of a number of binary hidden (latent) variables (Hinton and Salakhutdinov 2006; Larochelle et al. 2007). A drawback of RBM models is their inadequacy in sequential data modeling, since their (undirected) latent variable architecture is not designed for capturing temporal dependencies in the modeled data. To resolve these issues, conditional RBMs (CRBMs) have been recently proposed as an extension of RBMs (Taylor, Hinton, and Roweis 2011). CRBMs are based on the consideration of a time-varying nature for RBM biases, which are assumed to depend on the values of the previously observed data, in the context of an autoregressive data modeling scheme. Specifically, temporal dependencies are modeled by treating the observable variables in the previous time points as additional fixed inputs. This is effected by means of linear autoregressive connections from the past N configurations (time steps) of the observable variables to the current observable and hidden configuration.

On the other hand, echo-state networks (ESNs) are an efficient network structure for recurrent neural network (RNN) training (Lukosevicius and Jaeger 2009; Verstraeten et al. 2007; Maass, Natschlaeger, and Markram 2002). ESNs avoid the shortcomings of typical, gradient-descent-based RNN training, which suffers from slow convergence combined with bifurcations and suboptimal estimates of the model parameters (local optima of the optimized objective functions) (Haykin and Principe 1998; Kianifardand and Swallow 1996). This is accomplished by setting up the network structure in the following way:

- A recurrent neural network is randomly created and remains unchanged during training. This RNN is called the *reservoir*. It is passively excited by the input signal and maintains in its state a nonlinear transformation of the input history. The reservoir is not trained, but only initialized in a random fashion that ensures satisfaction of some constraints.
- The desired output signal is generated by a linear *readout* layer attached to the reservoir, which computes a linear combination of the neuron outputs from the input-excited reservoir *(reservoir states)*.

The updates of the reservoir state vectors and network outputs are computed as follows:

$$\boldsymbol{\phi}_{t+1} = (1-\gamma) h(\boldsymbol{\Lambda}\boldsymbol{\phi}_t + \boldsymbol{\Lambda}_{in}\boldsymbol{x}_{t+1}) + \gamma\boldsymbol{\phi}_t \quad (1)$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{\Lambda}_{readout}[\boldsymbol{x}_{t+1}; \boldsymbol{\phi}_{t+1}] \quad (2)$$

where $\boldsymbol{\phi}_t$ is the reservoir state at time t, $\boldsymbol{\Lambda}$ is the reservoir weight matrix, that is the matrix of the weights of the synaptic connections between the reservoir neurons, $\boldsymbol{x}_t$ is the observed signal fed to the network at time t, $\boldsymbol{y}_t$ is the obtained value of the readout at time t, $\gamma \geqslant 0$ is the *retainment rate* of the reservoir (with $\gamma > 0$ if leaky integrator neurons are considered), $\boldsymbol{\Lambda}_{readout}$ is the (linear) readout weights matrix, $\boldsymbol{\Lambda}_{in}$ are the weights between the inputs and the reservoir neurons, and $h(\cdot)$ is the activation function of the reservoir. *The parameters $\boldsymbol{\Lambda}_{in}$ and $\boldsymbol{\Lambda}$ of the network are not trained but only properly initialized* (Lukosevicius and Jaeger 2009).

Inspired from these advances, in this paper we propose a novel CRBM formulation that utilizes the merits of ESN reservoirs to capture complex nonlinear temporal dynamics in the modeled sequential data with increased modeling effectiveness, while entailing considerably less *trainable* model parameters. Training of the proposed model is conducted in an efficient way by means of contrastive divergence (CD) (Bengio and Delalleau 2008; Hinton 2002), while exact inference is possible in an elegant and computationally inexpensive way, similar to conventional CRBMs. We dub our approach the echo-state CRBM (ES-CRBM).

Further, we propose an implicit mixture of ES-CRBM experts (im-ES-CRBM), to incorporate in our model additional information regarding the allocation of the observed data to latent or observable classes, and effectively capture the transitions between such classes in the observed sequences. This allows for both obtaining better data modeling performance using our framework, as well as using our methods to perform sequential data classification (sequence segmentation). As we experimentally demonstrate, our methods outperform alternative RBM-based approaches, as well as other state-of-the-art methods, such as CRFs, in both data modeling and classification applications from diverse domains.

Proposed Approach

Echo-State Conditional RBM

Let us consider a sequence of observations $\{\boldsymbol{x}_t\}_{t=1}^{T}$. Let us also consider that each observation $\boldsymbol{x}_t$ is associated with a vector of hidden variables $\boldsymbol{h}_t$. Under the proposed ES-CRBM model, the joint density of the modeled observed and hidden variables yields:

$$p(\boldsymbol{x}_t, \boldsymbol{h}_t | \boldsymbol{x}_{<t}) = \frac{\exp(-E(\boldsymbol{x}_t, \boldsymbol{h}_t | \boldsymbol{x}_{<t}))}{\mathcal{Z}(\boldsymbol{x}_{<t})} \tag{3}$$

with the energy function of the model defined as

$$E(\boldsymbol{x}_t, \boldsymbol{h}_t | \boldsymbol{x}_{<t}) \triangleq -\sum_{ij} W_{ij} x_{it} h_{jt} - \sum_{i} \hat{a}_{it} x_{it} - \sum_{j} \hat{b}_{jt} h_{jt} \tag{4}$$

if we consider binary observations, or

$$E(\boldsymbol{x}_t, \boldsymbol{h}_t | \boldsymbol{x}_{<t}) \triangleq -\sum_{ij} W_{ij} x_{it} h_{jt} + \frac{1}{2} \sum_{i} (x_{it} - \hat{a}_{it})^2 - \sum_{j} \hat{b}_{jt} h_{jt} \tag{5}$$

in case of real-valued observations, similar to existing CRBMs (Taylor, Hinton, and Roweis 2011). However, contrary to existing CRBM formulations, the dynamic biases of the observed and hidden variables of our model are defined as

$$\hat{a}_{it} = a_i + \boldsymbol{A}_i \boldsymbol{\phi}_t \tag{6}$$

and

$$\hat{b}_{jt} = b_j + \boldsymbol{B}_j \boldsymbol{\phi}_t \tag{7}$$

respectively, where $\boldsymbol{\phi}_t$ is the reservoir state vector corresponding to the observations $\{\boldsymbol{x}_\tau\}_{\tau=1}^{t}$ of a suitable ESN postulated to capture the temporal dynamics in the modeled observations, given by (1).

From the above definition, it follows that two are the key features of our approach:

(i) Our method is capable of modeling highly non-linear temporal dependencies in the observed data, by utilizing an ESN reservoir to capture and encode temporal dynamics in the form of a high-dimensional state vector. This is in contrast to existing approaches, which rely on linearly combining past observations to capture the temporal dynamics underlying the modeled data.

(ii) Existing CRBM formulations are capable of retaining information up to r-steps in the past, where r is the order of the postulated model. Increasing the order of the model implies an increase in the number of *trainable* model variables and the incurred computational costs, while it might also give rise to overfitting effects. On the contrary, our ESN-based formulation is capable of capturing long temporal dynamics without requiring an analogous increase in the *trainable* model parameters.

Training for the proposed ES-CRBM model can be still performed by means of CD (Taylor, Hinton, and Roweis 2011). Specifically, parameter updating consists in a simple gradient ascent algorithm with updates

$$\Delta W_{ij} \propto \epsilon \sum_{t} \left[\langle x_{it} h_{jt} \rangle_{\text{data}} - \langle x_{it} h_{jt} \rangle_{\text{recon}} \right] \tag{8}$$

$$\Delta \boldsymbol{A}_i \propto \epsilon \sum_{t} \left[\langle x_{it} \boldsymbol{\phi}_t \rangle_{\text{data}} - \langle x_{it} \boldsymbol{\phi}_t \rangle_{\text{recon}} \right] \tag{9}$$

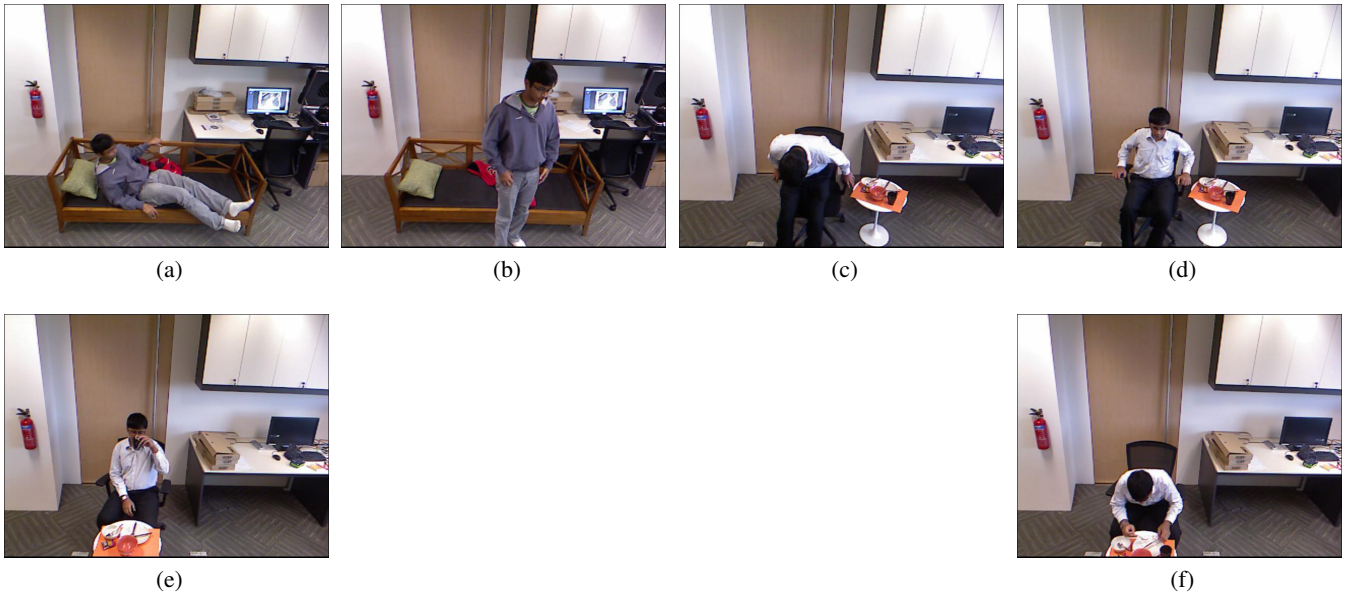
$$\Delta \boldsymbol{B}_j \propto \epsilon \sum_{t} \left[\langle h_{jt} \boldsymbol{\phi}_t \rangle_{\text{data}} - \langle h_{jt} \boldsymbol{\phi}_t \rangle_{\text{recon}} \right] \tag{10}$$

$$\Delta a_i \propto \epsilon \sum_{t} \left[\langle x_{it} \rangle_{\text{data}} - \langle x_{it} \rangle_{\text{recon}} \right] \tag{11}$$

$$\Delta b_j \propto \epsilon \sum_{t} \left[\langle h_{jt} \rangle_{\text{data}} - \langle h_{jt} \rangle_{\text{recon}} \right] \tag{12}$$

where, similar to simple RBMs, $\langle \cdot \rangle_{\text{data}}$ are the expectations with respect to the data distribution, and $\langle \cdot \rangle_{\text{recon}}$ are the expectations with respect to the k-step reconstruction distribution, obtained by *alternating Gibbs sampling*, starting with the observable variables clamped to the training data.

Data generation from a trained ES-CRBM can be performed by means of *alternating Gibbs sampling,* similar to simple RBMs. To obtain a joint sample from the CRBM distribution by means of *alternating Gibbs sampling*, we always keep the previous values of the observable variables fixed, pass these sequences through the employed ESN reservoir to obtain the corresponding reservoir state vectors $\boldsymbol{\phi}_t$, and pick new hidden and observable states that are compatible with each other and with the observable history, encoded in the state vectors $\boldsymbol{\phi}_t$. To start alternating Gibbs sampling, we typically initialize the value of the sought observable variable $\boldsymbol{x}_t$ at the value of $\boldsymbol{x}_{t-1}$ contaminated with some simple noise model (Taylor, Hinton, and Roweis 2011; Taylor et al. 2010).



(a) (b) (c) (d) (e) (f)

Figure 1: Depth image sequence segmentation experiments: Some characteristic frames.

Implicit Mixtures of Echo-State Conditional RBMs

Further, we extend our model to introduce an efficient mixture of ES-CRBM model experts. Such a model formulation allows for both performing classification of temporal observations (sequence segmentation) using our approach, as well as for incorporating into the modeling procedure the assumption that the observed sequential data belong to a number of primitive subpatterns, which may either be observable or not (latent variables).

Let us consider that the modeled observed data $\boldsymbol{x}$ are generated from K subpatterns (classes), which may comprise either latent or observable variables in our analysis. To exploit this information in the context of the data modeling mechanisms of the ES-CRBM, we modify it to incorporate the assumption that data pertaining to different classes correspond to different parameterizations of the observable/latent variable interaction patterns. Specifically, we define the energy function of our model in terms of three-way interactions among the observable variables $\boldsymbol{x}$, the latent variables $\boldsymbol{h}$, and the class variables (either latent or observable) $\boldsymbol{s}$, as follows:

$$E(\boldsymbol{x}_t, \boldsymbol{h}_t, \boldsymbol{s}_t|\boldsymbol{x}_{<t}) \triangleq -\sum_{ijc} W_{ij}^c x_{it} h_{jt} s_{ct} + \frac{1}{2}\sum_{ic}(x_{it} - \hat{a}_{it}^c)^2 s_{ct} - \sum_{jc} \hat{b}_{jt}^c h_{jt} s_{ct} \tag{13}$$

in case of continuous observed variables, and

$$E(\boldsymbol{x}_t, \boldsymbol{h}_t, \boldsymbol{s}_t|\boldsymbol{x}_{<t}) \triangleq -\sum_{ijc} W_{ij}^c x_{it} h_{jt} s_{ct} - \sum_{ic} \hat{a}_{it}^c x_{it} s_{ct} - \sum_{jc} \hat{b}_{jt}^c h_{jt} s_{ct} \tag{14}$$

in case of binary ones. In these equations, $\boldsymbol{s}_t$ is the class vector pertaining to observation $\boldsymbol{x}_t$; it comprises K entries, with the cth entry corresponding to the cth class. Of these entries, the entry that corresponds to the class generating $\boldsymbol{x}_t$ is equal to one, and the rest equal to zero.

From these assumptions, it follows that the energy function of our model yields:

$$p(\boldsymbol{x}_t, \boldsymbol{h}_t, \boldsymbol{s}_t|\boldsymbol{x}_{<t}) \propto \exp(-E(\boldsymbol{x}_t, \boldsymbol{h}_t, \boldsymbol{s}_t|\boldsymbol{x}_{<t})) \tag{15}$$

This way, the conditionals over the hidden variables $\boldsymbol{h}_t = (h_{jt})_{j=1}^J$, given the class variables, yield

$$p(h_{jt} = 1|\boldsymbol{x}_{\leq t}; s_{ct} = 1) = \frac{1}{1 + \exp\big(-\hat{b}_{jt}^c - \sum_i W_{ij}^c x_{it}\big)} \tag{16}$$

while for the observable variables $\boldsymbol{x}_t$ we have

$$p(x_{it}|\boldsymbol{h}_t, \boldsymbol{x}_{<t}; s_{ct} = 1) = \mathcal{N}(x_{it}|\hat{a}_{it}^c + \sum_j W_{ij}^c h_{jt}, 1) \tag{17}$$

in case of real-valued observations, and

$$p(x_{it} = 1|\boldsymbol{h}_t, \boldsymbol{x}_{<t}; s_{ct} = 1) = \frac{1}{1 + \exp\big(-\hat{a}_{it}^c - \sum_j W_{ij}^c h_{jt}\big)} \tag{18}$$

in case of binary observations. We dub the so-obtained model the implicit mixture of ES-CRBM (im-ES-CRBM) model.

Model Training. To train a C-component im-ES-CRBM model using a sequence of training observations, we resort to the familiar CD-k procedure. We consider two different model settings: (i) The case of unknown assignments of observations to classes; and, (ii) the case of known assignments of observations to classes.

We begin by considering the case of unknown assignments of observations to classes. In this case, the vectors

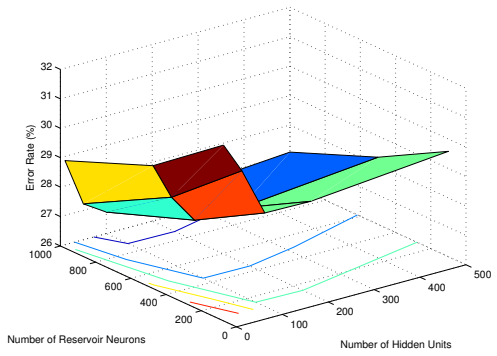


Figure 2: Depth image sequence segmentation experiments: im-ES-CRBM performance fluctuation with parameters.

$\boldsymbol{s}_t$ are considered latent variables. This scenario is implemented, e.g, in sequential data modeling applications such as trajectory tracking, where we may consider that complex trajectories comprise a set of alternating simpler latent motion primitives. Let us denote as $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$ a given training sequence; we perform model training using a set of training sequences X^m. Then, for each of the training sequences, the training algorithm for our model comprises the following steps:

Computation of latent class allocation (posterior) probabilities: At each time point t, we compute the probability of assignment of the observed vector $\boldsymbol{x}_t$ to each of the latent classes. For this purpose, we begin by noting that, from (15), it follows

$$p(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t}) \propto \exp(-F(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t})) \quad (19)$$

where the quantity $F(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t})$ is the *class-conditional free energy* of our model, and yields

$$F(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t}) = \frac{1}{2}\sum_i (x_{it} - \hat{a}_{it}^c)^2 - \sum_j \log\left(1 + \exp\left[\sum_i W_{ij}^c x_{it} + \hat{b}_{jt}^c\right]\right) \quad (20)$$

in case of continuous observations, and

$$F(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t}) = -\sum_i \hat{a}_{it}^c x_{it} - \sum_j \log\left(1 + \exp\left[\sum_i W_{ij}^c x_{it} + \hat{b}_{jt}^c\right]\right) \quad (21)$$

in case of binary observations.

Then, the (latent) class posteriors of our model straightforwardly yield (McLachlan and Peel 2000)

$$p(s_{ct} = 1|\boldsymbol{x}_{\leq t}) = \frac{\exp(-F(\boldsymbol{x}_t, s_{ct} = 1|\boldsymbol{x}_{<t}))}{\sum_{c'=1}^{C} \exp(-F(\boldsymbol{x}_t, s_{c't} = 1|\boldsymbol{x}_{<t}))} \quad (22)$$

Latent class sampling: Given the latent class posteriors $p(s_{ct} = 1|\boldsymbol{x}_{\leq t})$, we sample the assignments $\boldsymbol{s}_t^+$ of the observed vectors (henceforth denoted as) $\boldsymbol{x}_t^+$ to latent classes (and corresponding component ES-CRBM model).

ES-CRBM hidden variables sampling: We sample the hidden variables $\boldsymbol{h}_t$ pertaining to $\boldsymbol{x}_t^+$ from the $\boldsymbol{s}_t^+$th component ES-CRBM model (selected in the previous step). Let $\boldsymbol{h}_t^+$ be the resulting sampled value.

Reverse sampling of the observations: To realize the *alternating Gibbs sampling* procedure of CD-k, we subsequently resample (reconstruct) the observations $\boldsymbol{x}_t^+$ given the ES-CRBM model component selection $\boldsymbol{s}_t^+$ and the hidden variables sample $\boldsymbol{h}_t^+$. Let these (reconstructed) samples be denoted as $\boldsymbol{x}_t^-$.

Latent class sampling for the reconstructed data: We repeat the computation of the latent class posteriors for the reconstructed data, $\boldsymbol{x}_t^-$. Subsequently, we sample new assignments of the reconstructed data to the latent classes using the so-obtained posteriors of component assignment. Let $\boldsymbol{s}_t^-$ be the selected component.

Reverse sampling of the ES-CRBM latent variables: We sample the hidden variables $\boldsymbol{h}_t$ pertaining to $\boldsymbol{x}_t^-$ from the $\boldsymbol{s}_t^-$th component ES-CRBM model (selected in the previous step). Let $\boldsymbol{h}_t^-$ be the resulting sampled value.

Component ES-CRBM model parameters updating: Further, we update the parameters of the model component ES-CRBMs. For this purpose, we use the samples $\boldsymbol{x}^+$ and $\boldsymbol{h}^+$ assigned to each component ES-CRBM model to compute the related expectations with respect to the data distribution, and the final $\boldsymbol{x}^-$ and $\boldsymbol{h}^-$ samples assigned to each component ES-CRBM model to compute the related expectations with respect to the k-step reconstruction distribution. The obtained updating equations of the resulting gradient ascent algorithm yield

$$\Delta W_{ij}^c \propto \epsilon \sum \left[s_{ct}^+ x_{it}^+ h_{jt}^+ - s_{ct}^- x_{it}^- h_{jt}^-\right] \quad (23)$$

$$\Delta \boldsymbol{A}_i^c \propto \epsilon \sum \left[s_{ct}^+ x_{it}^+ \boldsymbol{\phi}_t - s_{ct}^- x_{it}^- \boldsymbol{\phi}_t\right] \quad (24)$$

$$\Delta \boldsymbol{B}_j^c \propto \epsilon \sum \left[s_{ct}^+ h_{jt}^+ \boldsymbol{\phi}_t - s_{ct}^- h_{jt}^- \boldsymbol{\phi}_t\right] \quad (25)$$

$$\Delta a_i^c \propto \epsilon \sum \left[s_{ct}^+ x_{it}^+ - s_{ct}^- x_{it}^-\right] \quad (26)$$

$$\Delta b_j^c \propto \epsilon \sum \left[s_{ct}^+ h_{jt}^+ - s_{ct}^- h_{jt}^-\right]. \quad (27)$$

Now, let us also consider the case of known assignments of observations to classes (component ES-CRBMs). This scenario is implemented, e.g, in sequential data classification applications, where the task is to segment a long sequence of observations into short segments corresponding to different (known) classes. In this case, we essentially repeat the previous learning algorithm, with the only difference being that we do not need to compute the posteriors of component assignment from (22) and subsequently sample the assignment variables, since these assignments are known. Instead, we replace them with the quantities

$$s_{ct} = \begin{cases} 1, & \text{if } \boldsymbol{x}_t \text{ is known to belong to } c\text{th component} \\ 0, & \text{otherwise} \end{cases}. \quad (28)$$

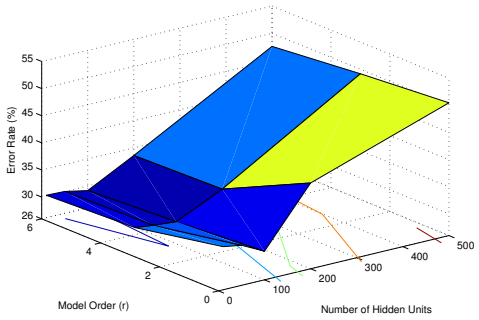


Figure 3: Depth image sequence segmentation experiments: imCRBM performance fluctuation with parameters.

Table 1: Depth image sequence segmentation experiments: Error rates obtained for optimal model configurations.

Method	Error Rate (%)	St.D.
5-CRF	27.13	0.013
LM	27.56	0.009
iSVM	30.41	0.011
imCRBM	29.15	0.018
im-ES-CRBM	26.53	0.014

This assignment rule applies to both the observed positive data and their corresponding (sampled) negative data.

Model Inference Algorithm. Model inference in the context of the im-ES-CRBM model comprises two distinct procedures: *data generation* and *data classification.*

Regarding data generation, we obtain a joint sample of $\{\boldsymbol{x}_t, \boldsymbol{h}_t\}$ by *alternating Gibbs sampling,* similar to standard RBM-type models. In this context, one iteration of the *alternating Gibbs sampling* algorithm comprises the following steps:

1. *Computation of the posterior distribution over the ES-CRBM model components given the known past observations,* $p(s_{ct} = 1|\boldsymbol{x}_{\leq t})$.
2. *Sampling of the model latent classes* $\boldsymbol{s}_t$ *corresponding to the observed data* $\boldsymbol{x}_t$. This is effected by utilizing the previously computed posteriors $p(s_{ct} = 1|\boldsymbol{x}_{\leq t})$.
3. *Sampling the latent variables of the component ES-CRBMs selected in step 2.* For this purpose, we use the conditionals $p(h_{jt} = 1|\boldsymbol{x}_{\leq t}; \boldsymbol{s}_t)$.
4. *Reconstruction of the observations.* For this purpose, we use the conditionals $p(x_{it}|\boldsymbol{h}_t, \boldsymbol{x}_{<t}; \boldsymbol{s}_t)$, and the previously sampled values of the $\boldsymbol{h}_t$ (latent variables) and $\boldsymbol{s}_t$ (emitting ES-CRBM components).

Regarding data classification, that is optimal assignment of the observed variables in a given sequence to the model's latent or observable classes, this can be effected by maximization of the posteriors $p(s_{ct} = 1|\boldsymbol{x}_{\leq t})$.

Experiments

In our experiments, we first consider application of im-ES-CRBM to sequential data classification, where each of the postulated model component ES-CRBMs corresponds to one known class in the modeled data; the task in this case is to find the correct assignment of the observations in the test sequences to the considered classes. Subsequently, we consider a data modeling application of both the ES-CRBM and im-ES-CRBM methods, dealing with trajectory-based robot learning by demonstration. In this case, we evaluate our models on the basis of the quality of the data generated from the modeled distribution. In all our experiments, the CD-k algorithm is performed with $k = 10$; all parameters use a gradient ascent learning rate equal to 10^{-3}, except for the autoregressive weights of the imCRBM method, where the learning rate is equal to 10^{-5}. A momentum term is also used: 0.9 of the previously accumulated gradient is added to the current gradient. We use hyperbolic-tangent *reservoir* neurons, $h(\cdot) \triangleq \tanh(\cdot)$; the reservoir spectral radius is set equal to 0.95.

Sequential data classification: Depth image sequence segmentation experiments

In this experiment, we evaluate our im-ES-CRBM method in segmenting and classifying depth image sequences, which depict humans performing actions in an assistive living environment. More specifically, our experiments are based on the dataset described in (Ni, Wang, and Moulin 2011). This dataset includes several actions from which we have selected the following: (1) get up from bed, (2) go to bed, (3) sit down, (4) eat meal, and (5) drink water. Some example frames from the considered dataset are depicted in Fig. 1. We seek to recognize these actions (1)-(5), using as our observable input the sequence of vectors $\boldsymbol{x}$ extracted as described next.

From this dataset, we extract features similar to (Ni, Wang, and Moulin 2011), by computing motion history images along the depth change directions. To calculate depth change, we use depth maps computed by a Kinect™ device. Kinect depth maps, however, contain a significant amount of noise. After frame differencing and thresholding, we noticed that motion was encoded even in areas with only still objects. To tackle this problem, we use median filtering. In the temporal domain, each pixel value is replaced by the minimum of its neighbors. Eventually, from these motion history images, we extract the first 12 complex Zernike coefficients (both norm and angle) (Kosmopoulos and Chatzis 2010), and use them as our feature vectors.

In our experiments, each action is contained in 35 video sequences. Each of these sequences, derived from the dataset presented in (Ni, Wang, and Moulin 2011), contains at least two of the considered actions (sequentially appearing). This setting enables us to assess the capacity of the evaluated algorithms to recognize these actions in real-world activities (in an assistive living environment). We subsample these video sequences by a factor of 2, similar to (Ni, Wang, and Moulin 2011). We use cross-validation in the following fashion: in each cycle, we use 15 randomly selected video se-

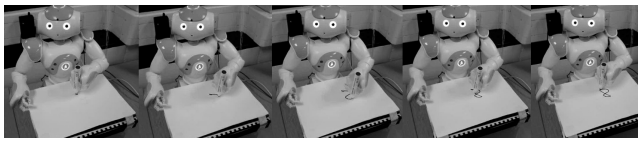


Figure 4: Robot learning by demonstration: NAO robot during the *Lazy figure 8* experiment.

quences to perform training, and keep the rest 20 for testing. We run the experiment 50 times to account for the effect of random selection of samples, and provide average performance measurements, and error bars. Recognition consists in assigning each feature vector to an action class.

Apart from the proposed approach, we also evaluate large-margin hidden Markov models (LM) (Sha and Saul 2007), moderate-order CRFs of 5th order (5-CRF) (Ye et al. 2009), the hidden Markov support vector machine (HMSVM) approach (Altun, Tsochantaridis, and Hofmann 2004), the iSVM approach (Zhu, Chen, and Xing 2011) with RBF kernels, and the imCRBM approach (Taylor et al. 2010). Both our approach and the imCRBM method are evaluated for various numbers of hidden variables (units), ranging from 10 to 500. We also try several values of the number of reservoir neurons for our method, ranging from 100 to $1,000$; we consider zero leaking rates for the reservoir neurons. In Fig. 2, we show how performance of our method changes with the number of hidden units and reservoir neurons. Further, in Fig. 3 we show how imCRBM model performance changes with the number of hidden units and model order. Finally, in Table 1 we demonstrate the optimal performance of our method and imCRBM (for optimal model configuration), as well as the performance obtained from the considered competitors.

As we observe, both our method and imCRBM yield their optimal performance when using 200 hidden units. However, while our model continues to improve its performance as we add reservoir neurons (i.e., extracting longer temporal dynamics from the modeled data), imCRBM performance seems to deteriorate for higher model orders. We also observe that our method works better than the considered state-of-the-art competitors. In contrast, imCRBM works better than iSVM only, i.e. a method not suitable for modeling data with temporal dynamics.

Sequential data modeling experiments: Trajectory-based robot learning by demonstration

Here, we examine the effectiveness of our approach in sequential data modeling. For this purpose, we consider a trajectory-based robot learning by demonstration experiment, adopted from our previous work (Chatzis and Demiris 2012). Specifically, we consider teaching a robot by demonstration how to draw a *lazy figure 8* (Fig. 4). The *lazy figure 8 (L8)* generation task is a classical benchmark for pattern generation methodologies (Chatzis and Demiris 2012). From the first impression, the task appears to be trivial, since an 8 figure can be interpreted as the superposition of a sine on the horizontal direction, and a cosine of half the sine's frequency on the vertical direction. A closer inspection though will reveal that in reality this seemingly easy task entails surprisingly challenging stability problems, which come to the fore when using limited model training datasets.

Table 2: Robot learning by demonstration: Obtained results (%) for optimal model configurations.

Method	Average	St.D.
PYP-GP	66.12	0.18
CRBM (500 Hidden Units, $r = 2$)	73.94	0.11
imCRBM (500 Hidden Units, $r = 4$)	77.40	0.13
ES-CRBM (500 Hidden / 100 reservoir neurons)	77.92	0.09
im-ES-CRBM (500 Hidden / 100 reservoir neurons)	86.31	0.04

In our experiments, we use the NAO robot (academic edition), a humanoid robot with 27 degrees of freedom (DoF). In these experiments, the modeled variable of the postulated models is the position vector of the robot joints. The captured dataset we use for our evaluations consists of joint angle data from drawing 3 consecutive *L8s*. The training trajectories are presented to the NAO robot by means of kinesthetics[1]; during this procedure, joint position sampling is conducted, with the sampling rate equal to 20 Hz. We use multiple demonstrations, so as to capture the variability of human action. Specifically, we have recorded 4 demonstrations and used 3 for training and 1 for testing purposes. Due to the temporal variations observed in the demonstrations, we have pre-processed the sequences using Dynamic Time Warping (DTW).

Performance assessment is performed on the basis of the *percentage of explained variance* (Bakker and Heskes 2003), defined as the total variance of the data minus the sum-squared error on the test set as a percentage of the total variance. Apart from our ES-CRBM and im-ES-CRBM approaches, the evaluated methods comprise CRBM (Taylor, Hinton, and Roweis 2011), imCRBM, and PYP-GP (Chatzis and Demiris 2012). imCRBM and im-ES-CRBM are trained in a *completely* unsupervised way, using only 2 mixture components in each case. We experiment with multiple numbers of hidden units, reservoir neurons, and model orders r (wherever applicable), and report results for optimal selection. We consider leaking rates equal to 0.9 for the reservoir neurons.

In Table 2, we provide the results (means and standard deviations obtained by application of leave-one-out cross-validation) of the evaluated methods. Our main findings are the following: (i) The proposed methods yield a clear competitive advantage over their CRBM and imCRBM counterparts. (ii) The implicit mixture CRBM variants seem to yield superior performance compared to their single-component counterparts. This performance improvement is much more

[1]Manually moving the robot's arms and recording the joint angles.

prominent in the case of the proposed models, compared to the conventional CRBM formulations. Finally, we also note the notable performance gains of all the considered energy-based models compared to the PYP-GP method, which relies on a much more computationally expensive Gaussian process model to perform sequential data modeling.

Conclusions

In this paper, we proposed a method exploiting the merits of ESNs to enhance the sequential data modeling capabilities of CRBMs. Our approach consists in the utilization of an ESN reservoir to capture the temporal dynamics in the context of CRBMs instead of the linear autoregressive apparatus of existing approaches. This model formulation allows for extracting more complex temporal dynamics using less *trainable* model parameters.

Subsequently, we extended the so-obtained ES-CRBM model to obtain an implicit mixture of ES-CRBM experts, capable of better modeling multimodal sequential data, as well as performing classification of observed sequences, apart from data modeling and generation. Exact inference for our models was performed by means of an elegant alternating Gibbs sampling algorithm, while training was conducted by means of CD.

Our future goals focus on investigating the efficacy of our approach in a wide class of applications involving time-series data or data with temporal dynamics. Characteristic application areas include dynamic planning algorithms for multirobot swarms, automatic music improvisation and metacreation, and analysis and prediction of asset prices in financial markets using high-frequency measurements.

References

Altun, Y.; Tsochantaridis, I.; and Hofmann, T. 2004. Hidden Markov support vector machines. In *Proc. ICML*.

Bakker, B., and Heskes, T. 2003. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research* 4:83–99.

Bengio, Y., and Delalleau, O. 2008. Justifying and generalizing contrastive divergence. *Neural Computation* 21(1):1–21.

Chatzis, S. P., and Demiris, Y. 2012. Nonparametric mixtures of Gaussian processes with power-law behavior. *IEEE Transactions on Neural Networks and Learning Systems* 23(12):1862–1871.

Haykin, S., and Principe, J. 1998. Making sense of a complex world. *IEEE Signal Process. Mag.* 15(3):66–81.

Hinton, G. E., and Salakhutdinov, R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504 – 507.

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

Kianifardand, F., and Swallow, W. 1996. A review of the development and application of recursive residuals in linear models. *J. Amer. Statist. Assoc.* 91(443):391–400.

Kosmopoulos, D., and Chatzis, S. 2010. Robust visual behavior recognition. *Signal Processing Magazine, IEEE* 27(5):34 –45.

Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; and Bengio, Y. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proc. ICML*, 473–480. ACM Press.

Lukosevicius, M., and Jaeger, H. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3:127–149.

Maass, W.; Natschlaeger, T.; and Markram, H. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14(11):2531–2560.

McLachlan, G., and Peel, D. 2000. *Finite Mixture Models*. New York: Wiley Series in Probability and Statistics.

Nair, V., and Hinton, G. 2008. Implicit mixtures of restricted Boltzmann machines. In *Proc. NIPS*.

Ni, B.; Wang, G.; and Moulin, P. 2011. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *ICCV Workshops*, 1147–1153.

Sha, F., and Saul, L. K. 2007. Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models. In *Proc. ICASSP*, 313–316.

Smolensky, P. 1986. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland, e. a., ed., *Parallel Distributed Processing: Volume 1: Foundations*, 194–281. Cambridge, MA: MIT Press.

Taylor, G. W.; Sigal, L.; Fleet, D. J.; and Hinton, G. E. 2010. Dynamical binary latent variable models for 3d human pose tracking. In *Proc. CVPR*, 631–638.

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

Verstraeten, D.; Schrauwen, B.; D'Haene, M.; and Stroobandt, D. 2007. 2007 special issue: An experimental unification of reservoir computing methods. *Neural Networks* 20(3):391–403.

Ye, N.; Lee, W. S.; Chieu, H. L.; and Wu, D. 2009. Conditional random fields with high-order features for sequence labeling. In *Proc. NIPS*.

Zhu, J.; Chen, N.; and Xing, E. P. 2011. Infinite SVM: a Dirichlet process mixture of large-margin kernel machines. In *Proc. ICML*.