AIMS Mathematics, 11(1): 272–290.
DOI: 10.3934/math.2026011
Received: 26 September 2025
Revised: 03 December 2025
Accepted: 23 December 2025
Published: 05 January 2026

https://www.aimspress.com/journal/Math

Research article

Deep Bayesian networks

Michael A. Kouritzin*

Department of Mathematical Sciences, University of Alberta, Edmonton, Alberta, Canada

* **Correspondence:** Email: michaelk@ualberta.ca.

Abstract: Deep Bayesian networks (DBNs) having deep recurrent neural network (DRNN) topography, effective forward-backward learning and innate model selection capabilities are introduced. DBNs provided randomness, Bayes' factor methods and efficient gradient-free expectation-maximization-based (EM) learning to the DRNN layout. DBN's learning, simulation and Bayes' factor capabilities provided an effective generative adversarial network (GAN) in the sequential (RNN) setting. Consequently, deep fakes with real probabilistic models could be created, based upon training data. Alternatively, DBNs could be thought of as some super generalization of hidden Markov models (HMMs), which have inputs and multiple hidden layers. The proofs establishing the above claims were based upon the novel idea to transform the whole network to a completely independent network where the analysis is trivial using a Girsanov like theorem.

Keywords: recurrent neural network; hidden Markov model; model selection; big data; Girsanov's theorem; machine learning
Mathematics Subject Classification: 62M05, 60J22, 68T10

1. Introduction

Feedforward neural networks (NNs) resemble input-output mappings while recurrent neural networks (RNNs) allow feedback. RNNs model phase transitions (using Hopfield type RNNs) or handle sequential data (using Deep RNNs), like time series or natural language data, by maintaining an internal memory of previous inputs. RNNs can be traced back to the Ising model created and analyzed by [7,11], which is a nearest-neighbor connected RNN with the nodes only taking two values. [1] made this architecture adaptive and [5] generalized and popularized the learning of this class of RNN, which have become known as Hopfields.

Deep recurrent neural network (DRNN) are generally non-thresholded types of RNNs that learn to recognize sequential data patterns. They have input, hidden and output layers like feedforward NNs but can take the output from one layer of recurrent unit and feed it into the next recurrent unit (see

Figure 1), allowing the network to capture more complex relationships between the input and output sequences. The number of layers in a deep RNN can vary depending on the complexity of the problem being solved, and the number of hidden units in each layer can also be adjusted. Recently, transformers have been viewed as infinitely deep RNN (see [13]).

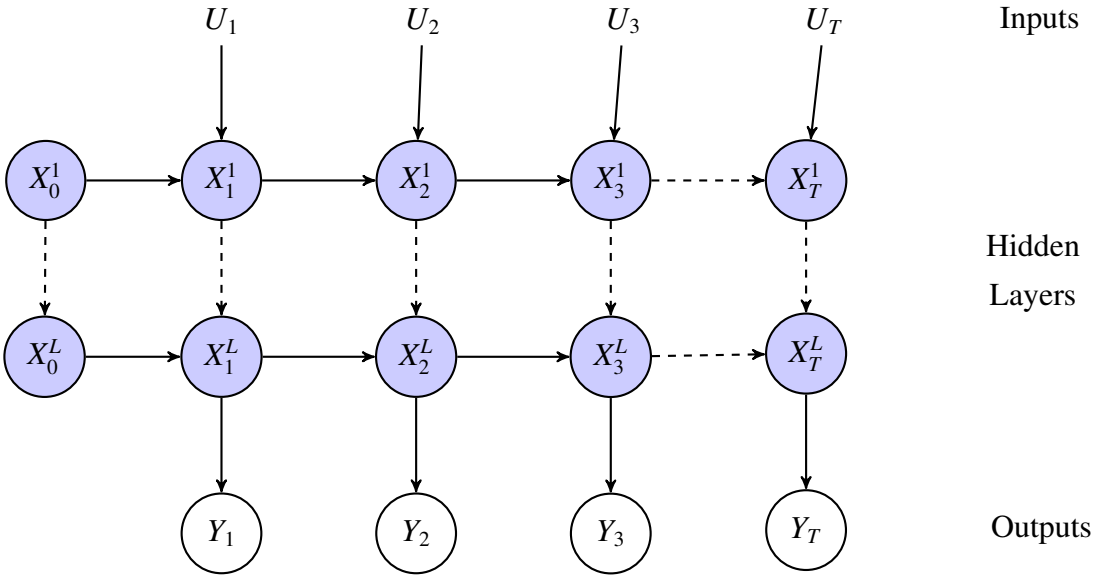

Figure 1. Deep recurrent neural network expanded in time.

We look at the DRRN a little closer because the model we introduce in this paper shares a similar structure. As shown in Figure 1, the current output usually does not depend upon prior ones. Indeed, if W's represent weight matrices, b's bias vectors and σ's represent different activation functions, then the DRNN equations are:

$$\begin{aligned} X_t^1 &= \sigma^1(W_U^1 \cdot U_t + W_X^1 \cdot X_{t-1}^1 + b^1), \\ X_t^2 &= \sigma^2(W_1^2 \cdot X_t^1 + W_X^2 \cdot X_{t-1}^2 + b^2), \\ &\vdots \\ X_t^L &= \sigma^L(W_1^L \cdot X_t^{L-1} + W_X^L \cdot X_{t-1}^L + b^L), \\ Y_t &= W^Y \cdot X_t^L + b^Y, \end{aligned}$$

where $W \cdot x$ denotes matrix multiplication and each scalar activation function σ is applied separately to each component. For notational convenience, we let $X_t^0 = U_t$ for all t, N^l be the dimension of the l layer, so W_1^i, W_X^i are $N^i \times N^{i-1}$ while b^i is N^i, and θ denote the collection of weights W and biases b to be learned. The network can be made arbitrarily deep by choosing a large enough L. The *universal approximation* theorem of [14] says that *any* open dynamical system with continuous coefficient can be approximated arbitrarily well by a DRNN with $L = 1$ and any T by letting $N^1 \to \infty$.

The system parameters Θ (the W's and b's here) are normally learned by some adaptive stochastic gradient method like ADAM but these suffer from the vanishing gradient issue (see [6]) when L and/or T gets large. In general, DRNNs have the advantages of allowing inputs, multiple hidden layers, universal approximation of dynamical systems and highly refined computer packages. They have the

general weaknesses of having no observation dependence, being completely deterministic, having no model selection and being difficult to learn due to the vanishing/exploding gradient issue. Still, DRNNs are particularly useful in applications where there is a lot of sequential data to process, like speech recognition (Amazon's Alexa, Apple's Siri), natural language processing (Google Translate), music generation (Magenta's MusicVAE), autonomous driving (Waymo), intrusion detection, fraud detection, protein function prediction, ECG signal denoising, and human activity recognition. Moreover, tranformers, now considered infinitely deep recurrent networks, are used in Captionbots (Google's Show and Tell) and Chatbots (ChatGPT). Long-short-term-memory (LSTM) networks are used as partial fix to the exploding gradient issue of DRNN but there is no known universal approximation to dynamical systems for LSTMs. This paper addresses two of main difficulties that arise in using DRNN, choosing the exact network topography and the vanishing gradient problem when training large networks, by introducing an alternative (stochastic) network that has a more general layout.

The purpose of this paper is to develop a probabilistic deep recurrent neural network, termed the deep Bayesian network (DBN), which is parameterized in a different manner. This DBN will generalize the Markov observation model introduced in [9] and thereby also generalize the hidden Markov model. Each layer, including the input and output, is a Markov chain that depends on the previous layer. In the case of the first hidden layer, the Markov chain dependence is on the input, which can be a control based upon prior outputs. More precisely,

Definition 1.1. *Suppose* $(\Omega, \mathcal{F}, P)$ *is some probability space. Then,* $\{(U_n, X_n^1, X_n^2, ..., X_n^L, Y_n), n \geq 0\}$ *is a* (r, p, q, μ) *DBN on* $(\Omega, \mathcal{F}, P)$ *if:*

(1) The input $\{U_n, n \geq 0\}$ *is a conditional* $r_{u,\widehat{u}}(y)$*-Markov chain whose transition probabilities can depend upon the prior output (if any) i.e.*

$$P(U_n = \widehat{u} | U_{n-1} = u, Y_{n-1} = y; \{X_j^l\}_{j<n}^{l\leq L}, \{U_j\}_{j<n-1}, \{Y_j\}_{j<n-1}) = r_{u,\widehat{u}}(y);$$

(2) The first hidden layer $\{X_n^1, n \geq 1\}$ *is a conditional* $p_{x,\widehat{x}}^1(\widehat{u})$*-Markov chain whose transition probabilities depend upon the input (if any) i.e.*

$$P(X_n^1 = \widehat{x} | X_{n-1}^1 = x, U_n = \widehat{u}; \{X_j^l\}_{j<n}^{l\leq L}, \{U_j\}_{j<n}, \{Y_j\}_{j<n}) = p_{x,\widehat{x}}^1(\widehat{u});$$

(3) The i^{th} *hidden layer* $\{X_n^i, n \geq 1\}$*,* $1 < i \leq L$*, is a conditional* $p_{x,\widehat{x}}^i(\widehat{x^{i-1}})$*-Markov chain whose transition probabilities depend upon the prior layer i.e.*

$$P(X_n^i = \widehat{x} | X_{n-1}^i = x, X_n^{i-1} = \widehat{x^{i-1}}; \{X_n^l\}^{l<i-1}, \{X_j^l\}_{j<n}^{l\leq L}, \{U_j\}_{j\leq n}, \{Y_j\}_{j<n}) = p_{x,\widehat{x}}^i(\widehat{x^{i-1}});$$

(4) The output layer $\{Y_n, n \geq 1\}$ *is a conditional* $q_{y,\widehat{y}}(\widehat{x^L})$*-Markov chain whose transition probabilities depend upon the last hidden layer i.e.*

$$P(Y_n = \widehat{y} | Y_{n-1} = y, X_n^L = \widehat{x^L}; \{X_n^l\}^{l<L}, \{X_j^l\}_{j<n}^{l\leq L}, \{U_j\}_{j\leq n}, \{Y_j\}_{j<n-1}) = q_{y,\widehat{y}}(\widehat{x^L});$$

(5) The joint initial distribution satisfies $\mu(A) = P(U_0, X_0^1, ..., X_0^L, Y_0 \in A)$ *for Borel A*

for all $n, u, \widehat{u}, x, \widehat{u}, y, \widehat{y}$*.*

Remark 1.2. *In (3), there is double restriction on* X_{n-1}^i *through* $X_{n-1}^i = x$ *and* $\{X_n^l\}_{j<n}^{l\leq L}$ *to avoid overly complicating the notation. The second one becomes meaningless for the* i^{th}*-layer in presence of the first one* $X_{n-1}^i = x$*. A similar notation simplification is done in (2).*

Remark 1.3. *We set $X_j^0 = U_j$, $X_j^{L+1} = Y_j$ for all j for notational ease.*

These conditions collectively establish that the hidden layers are Markov given the past inputs and observations. Indeed, it follows from the multiplication rule and the previous definition that for $j \in \{1, ..., L\}$

$$\begin{aligned} & P\left(\left\{X_{n+1}^l = \widehat{x}^l\right\}_{l=1}^{j} \mid U_{n+1} = \widehat{x}^0, \left\{X_n^l = x^l\right\}_{l=1}^{j}, \left\{X_j^l\right\}_{j\le n-1}^{l\le L}, \left\{U_j\right\}_{j\le n}, \left\{Y_j\right\}_{j\le n}\right) \\ = \; & P\left(X_{n+1}^j = \widehat{x}^j \middle| \left\{X_{n+1}^l = \widehat{x}^l\right\}_{l=1}^{j-1}, U_{n+1} = \widehat{x}^0, \left\{X_n^l = x^l\right\}_{l=1}^{j}, \left\{X_j^l\right\}_{j\le n-1}^{l\le L}, \left\{U_j\right\}_{j\le n}, \left\{Y_j\right\}_{j\le n}\right) \\ \times \; & P\left(X_{n+1}^{j-1} = \widehat{x}^{j-1} \middle| \left\{X_{n+1}^l = \widehat{x}^l\right\}_{l=1}^{j-2}, U_{n+1} = \widehat{x}^0, \left\{X_n^l = x^l\right\}_{l=1}^{j}, \left\{X_j^l\right\}_{j\le n-1}^{l\le L}, \left\{U_j\right\}_{j\le n}, \left\{Y_j\right\}_{j\le n}\right) \\ & \times \cdots \times P\left(X_{n+1}^1 = \widehat{x}^1 \mid U_{n+1} = \widehat{x}^0, \left\{X_n^l = x^l\right\}_{l=1}^{j}, \left\{X_j^l\right\}_{j\le n-1}^{l\le L}, \left\{U_j\right\}_{j\le n}, \left\{Y_j\right\}_{j\le n}\right) \\ = \; & \prod_{l=1}^{j} p_{x^l,\widehat{x}^l}^l\left(\widehat{x}^{l-1}\right), \end{aligned} \tag{1.1}$$

where $\widehat{x}^0 = u$. In a similar way, one can show $\left\{U_j, \{X_j^l\}_{l=1}^L, Y_j\right\}_{j\le n}$ is Markov

$$\begin{aligned} & P\left(U_{n+1} = \widehat{u}, \left\{X_{n+1}^l = \widehat{x}^l\right\}, Y_{n+1} = \widehat{y} \middle| U_n = u, \left\{X_n^l = x^l\right\}, Y_n = y; \left\{U_j, \left\{X_j^l\right\}, Y_j\right\}_{j<n}\right) \\ = \; & P\left(Y_{n+1} = \widehat{y} \middle| Y_n = y, \left\{X_{n+1}^l = \widehat{x}^l\right\}, U_{n+1} = \widehat{u}, U_n = u, \left\{X_n^l = x^l\right\}; \left\{U_j, \left\{X_j^l\right\}, Y_j\right\}_{j<n}\right) \\ & \times P\left(\left\{X_{n+1}^l = \widehat{x}^l\right\} \middle| \left\{X_n^l = x^l\right\}, U_{n+1} = \widehat{u}, Y_n = y, U_n = u; \left\{U_j, \left\{X_j^l\right\}, Y_j\right\}_{j<n}\right) \\ & \times P\left(U_{n+1} = \widehat{u} \middle| U_n = u, Y_n = y, \left\{X_n^l = x^l\right\}; \left\{U_j, \left\{X_j^l\right\}, Y_j\right\}_{j<n}\right) \\ = \; & q_{y,\widehat{y}}(\widehat{x}^L)\left[\prod_{l=1}^{L} p_{x^l,\widehat{x}^l}^l\left(\widehat{x}^{l-1}\right)\right] r_{u,\widehat{u}}(y), \end{aligned} \tag{1.2}$$

where the last line is the transition probability for $\left\{U_j, \{X_j^l\}_{l=1}^L, Y_j\right\}_{j\le n}$.

The weight and bias parameters of the usual DRNN are replaced with the new model parameters θ of μ, $r_{u,\widehat{u}}(y)$, $p_{x,\widehat{x}}^i(\widehat{x}^{i-1})$ all i, $q_{y,\widehat{y}}(\widehat{x}^L)$ for DBN. This is why we also refer to DBN as a reparameterized DRNN. With these new parameters we will show that there is a Baum-Welch-like expectation-maximization algorithm that is computationaly stable for learning these new parameters. In particular, there is no (vanishing) gradient typical for RNN learning. As it is a probabilistic alternative, DBN is a natural network for representing random time-series, stochastic differential equations or other Markov phenomena. DBN can be used as DRNN but also for such things as learning stochastic differential equations or producing deep fakes backed by known stochastic models. The main advantages of this model over DRNN are: It allows input and observation dependence as shown in Figure 2, model selection and model averaging, Baum-Welch-like EM learning and Viterbi maximally likely hidden sequence determination.

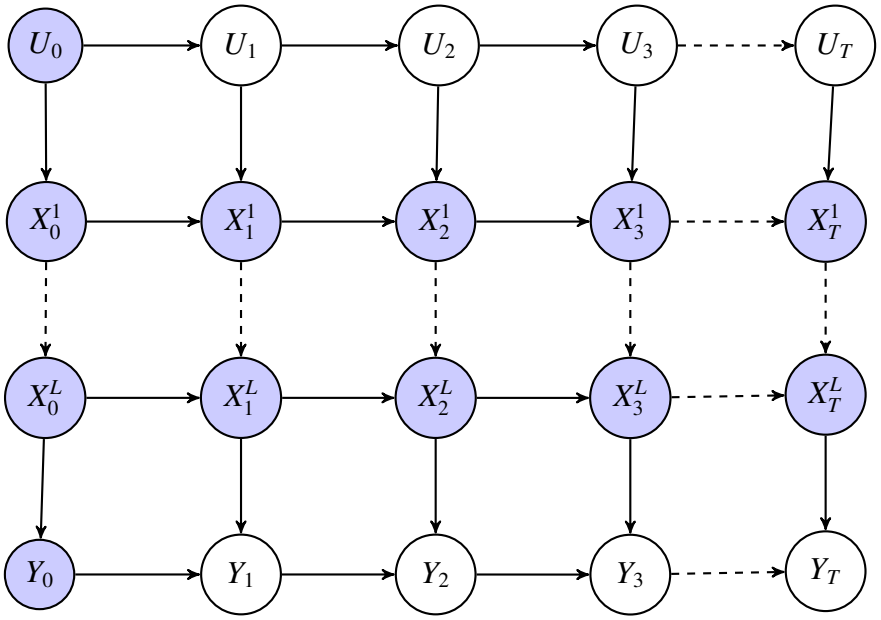

Figure 2. Deep Markov observation model.

The point of this paper is to introduce a new model and the mathematics behind it. There are obviously many possible interesting applications. For example, this work extends the models in [9] so deepfake applications in that paper apply to our present work. Furthermore, these models seem to be able to replace particle filters (like those in [2, 10]) so applications like those in [3, 4, 12] should apply as well as many others. However, proper investigation of applications will be left to future works so as not to lose focus.

2. Reference space

Motivated in part by [8], our approach is to construct the real or target DBN model through measure transform, starting from a canonical, overly simple, proposal model for $\{(U_n, X_n^1, X_n^2, ..., X_n^L, Y_n), n \geq 0\}$. The proposal model existence is obvious and it is created on a measurable space $(\Omega, \mathcal{F})$ with a reference measure $\overline{P}$. With respect to this reference measure, we assume that:

(1) The input $\{U_n, n \geq 0\}$ is a conditional $r_{u,\widehat{u}}(y)$-Markov chain whose transition probabilities can depend upon the prior output (if any). Hence, the input model is the same under the proposal and DBN models. This is because both the input and output sequences are known during training and the input model just depends upon these two sequences;
(2) $\{X_n^1, n \geq 1\}$ is a $\overline{p}_{x,\widehat{x}}^1$-Markov chain. It does not depend upon input nor output;
(3) Each $\{X_n^i, n \geq 1\}$ is a $\overline{p}_{x,\widehat{x}}^i$-Markov chain. It does not depend upon prior hidden layers;
(4) $\{Y_n, n \geq 1\}$ is a $\overline{q}_{y,\widehat{y}}$-Markov chain. It does not depend upon the hidden layers;
(5) $\overline{P}(U_0 = u_0, X_0^1 = x_1, ..., X_0^L = x_L, Y_0 = y_0) = \overline{\mu}_1(x_1) \cdots \overline{\mu}_L(x_L)\overline{\mu}_{UY}(u_0, y_0)$ for some probability measures $\overline{\mu}_1, ..., \overline{\mu}_L, \overline{\mu}_{UY}$.

The calculations are trivial for the proposal model as we can treat each layer on its own with the exception that the input transitions can still depend upon the output. The proposal model clearly exists and is easy to simulate. We turn to changing the proposal into the target.

Let $\mathcal{F}_t = \sigma\{U_s, X_s^1, ..., X_s^L, Y_s; s \le t, U_{t+1}\}$ be the information up to t and define the likelihood ratio

$$A_t = \frac{\mu\left(U_0, X_0^1, .., X_0^L, Y_0\right)}{\overline{\mu}_1\left(X_0^1\right)\cdots\overline{\mu}_L\left(X_0^L\right)\overline{\mu}_{UY}\left(U_0, Y_0\right)} \prod_{n=1}^{t} \frac{q_{Y_{n-1},Y_n}\left(X_n^L\right)}{\overline{q}_{Y_{n-1},Y_n}} \prod_{i=1}^{L} \frac{p_{X_{n-1}^i,X_n^i}(X_n^{i-1})}{\overline{p}_{X_{n-1}^i,X_n^i}}. \tag{2.1}$$

Let $\overline{E}$ denote (conditional) expectation using probability measure $\overline{P}$. Then, the likelihood A defined above will turn the proposal model into the target DBN model.

Lemma 2.1. *$\{A_t, t \ge 0\}$ is a $\{\mathcal{F}_t\}_{t\ge 0}$-martingale with respect to $\overline{P}$. If $\{(U_t, \{X_t^l\}_{l\le L}, Y_t), t \in \{0, 1, 2, ..., T\}\}$ is the proposal model defined above in (1)–(5) with respect to $\overline{P}$, then it becomes the target (r, p, q, μ) DBN model with respect on $(\Omega, \mathcal{F}_T, P)$, where $\frac{dP}{d\overline{P}} = A_T$ on $\mathcal{F}_T$.*

Proof. The martingale property follows from the tower property and

$$\begin{aligned}
\overline{E}\left[A_{t+1} \mid \mathcal{F}_t\right] &= A_t \overline{E}\left[\frac{q_{Y_t,Y_{t+1}}\left(X_{t+1}^L\right)}{\overline{q}_{Y_t,Y_{t+1}}}\left[\prod_{i=2}^{L}\frac{p_{X_t^i,X_{t+1}^i}(X_{t+1}^{i-1})}{\overline{p}_{X_t^i,X_{t+1}^i}}\right]\frac{p_{X_t^1,X_{t+1}^1}(U_{t+1})}{\overline{p}_{X_t^i,X_{t+1}^i}} \mid \mathcal{F}_t\right] \\
&= A_t \sum_{x_{t+1}^1}\cdots\sum_{x_{t+1}^L}\sum_{y_{t+1}} q_{Y_t,y_{t+1}}\left(x_{t+1}^L\right)\left[\prod_{i=2}^{L} p_{X_t^i,x_{t+1}^i}(x_{t+1}^{i-1})\right] p_{X_t^1,x_{t+1}^1}(U_{t+1}) \\
&= A_t,
\end{aligned} \tag{2.2}$$

using the previsibility of U with $\{\mathcal{F}_t\}$ defined above and summing from the inside out. Next, using the Bayes' rule lemma to follow immediately (with $s = t$ and $\mathcal{G}_s = \mathcal{F}_s$) and (2.2), one has that

$$\begin{aligned}
&P\left(U_{t+1} = u, \{X_{t+1}^i = x^i\}_{i=1}^L, Y_{t+1} = y \mid \mathcal{F}_t\right) \\
&= \frac{\overline{E}\left[A_{t+1}1_{U_{t+1}=u,\{X_{t+1}^i=x^i\},Y_{t+1}=y} \mid \mathcal{F}_t\right]}{\overline{E}\left[A_{t+1} \mid \mathcal{F}_t\right]} \\
&= \frac{A_t\overline{E}\left[\frac{q_{Y_t,y}(x^L)}{\overline{q}_{Y_t,y}}\left[\prod_{i=2}^{L}\frac{p_{X_t^i,x^i}(x^{i-1})}{\overline{p}_{X_t^i,x^i}}\right]\frac{p_{X_t^1,x^1}(u)}{\overline{p}_{X_t^i,x^i}}1_{U_{t+1}=u,\{X_{t+1}^i=x^i\},Y_{t+1}=y} \mid \mathcal{F}_t\right]}{A_t} \\
&= q_{Y_t,y}\left(x^L\right)\left[\prod_{i=2}^{L} p_{X_t^i,x^i}(x^{i-1})\right] p_{X_t^1,x^1}(u) r_{U_t,u}(Y_t),
\end{aligned} \tag{2.3}$$

so the DBN model has been achieved by measure change. □

We just used the following Bayes' rule with $\mathcal{G}_t = \mathcal{F}_t$. Later, we will use it with $\mathcal{G}_t = \sigma\{Y_1, ..., Y_t\}$ when concerned about the filter.

Lemma 2.2 (Bayes)**.** *Let $\frac{dP}{d\overline{P}} = A_T$ on $\mathcal{F}_T$, $\mathcal{G}_s \subset \mathcal{F}_s$, Z_t be a bounded $\mathcal{F}_t$ random variable and $s \le t \le T$. Then, $E\left[Z_t \middle| \mathcal{G}_s\right] = \frac{\overline{E}\left[A_t Z_t \middle| \mathcal{G}_s\right]}{\overline{E}\left[A_s \middle| \mathcal{G}_s\right]}$.*

Proof. Cross multiplying and using martingale property, one needs to show

$$\overline{E}\left[E\left[Z_t\middle|\mathcal{G}_s\right]\overline{E}\left[A_s\middle|\mathcal{G}_s\right]1_F\right] = \overline{E}\left[A_t Z_t 1_F\right] = \overline{E}\left[A_T Z_t 1_F\right]$$

for all $F \in \mathcal{G}_s$. However, by martingale property for A

$$\overline{E}\left[A_s\middle|\mathcal{G}_s\right] = \overline{E}\left[\overline{E}\left[A_T\middle|\mathcal{F}_s\right]\middle|\mathcal{G}_s\right] = \overline{E}\left[A_T\middle|\mathcal{G}_s\right]$$

and then taking out knowns and finally total expectation

$$\begin{aligned}
\overline{E}\left[E\left[Z_t\middle|\mathcal{G}_s\right]\overline{E}\left[A_T\middle|\mathcal{G}_s\right]1_F\right] &= \overline{E}\left[\overline{E}[A_T E\left[Z_t\middle|\mathcal{G}_s\right]1_F|\mathcal{G}_s]\right] \\
&= \overline{E}\left[A_T E\left[Z_t\middle|\mathcal{G}_s\right]1_F\right] \\
&= E\left[E\left[Z_t\middle|\mathcal{G}_s\right]1_F\right] \\
&= E\left[E\left[Z_t 1_F\middle|\mathcal{G}_s\right]\right] \\
&= E\left[Z_t 1_F\right] \\
&= \overline{E}\left[A_T Z_t 1_F\right]
\end{aligned}$$

for all $F \in \mathcal{G}_s$. □

3. Forward ratio, Filter and Bayes' factor

In this section, we will introduce the filter for estimating the hidden states based upon back inputs and outputs. This will be formed in terms of a particular unnormalized filter, which is formed from the forward ratio that has a nice recursion. The total mass of this forward ratio also gives the Bayes' factor of the DBN model under question over the canonical proposal model. To ease the notation in the sequel, we will let

$$U_{i,j} = \begin{cases} (U_i, ..., U_j) & j \geq i \\ 1 & j < i \end{cases}$$

and similar for outputs Y as well as other variables.

The filter, π, is the conditional distribution of the hidden states given the inputs and outputs up to the current time. Specifically, it is defined as the measure-valued process:

$$\pi_t(\{x^l\}_{l=1}^L) = P\left(\{X_t^l\} = \{x^l\} \mid U_{1,t}, Y_{1,t}\right) \tag{3.1}$$

for $t = 0, 1, ..., T$. The filter is often found by normalizing an *unnormalized filter*. Such an unnormalized filter can be formed by shifting the *forward ratio*, which is defined as:

$$\alpha_t(\{\widehat{x}^l\}_{l=1}^L) = \frac{P(U_{1,t-1}, Y_{1,t-1}, \{X_{t-1}^l = \widehat{x}^l\}_{l=1}^L \mid \theta)}{\overline{P}(U_{2,t-1}, Y_{2,t-1}|U_1, Y_1)} \tag{3.2}$$

for $t = 2, ..., T, T+1$ with

$$\alpha_1(\{\widehat{x}^l\}_{l=1}^L, u, y) = P(\{X_0^l = \widehat{x}^l\}_{l=1}^L, U_0 = u, Y_0 = y \mid \theta) = \mu(u, \{\widehat{x}^l\}_{l=1}^L, y). \tag{3.3}$$

α is also important for determining parameters θ through the EM algorithm and satisfies a very convenient forward recursion that will help make this EM algorithm efficient. Indeed, it follows by partitioning, the multiplication rule, (3.2, 3.3) as well as the Markov properties with P and $\overline{P}$ that

$$
\begin{aligned}
&\alpha_2(\{\widehat{x}^l\}) \\
&= \sum_{u;\{x^l\};y} P(\{X_0^l = x^l\}, U_0 = u, Y_0 = y) P(\{X_1^l = \widehat{x}^l\}, U_1, Y_1 \,|\{X_0^l = x^l\}, U_0 = u, Y_0 = y) \\
&= \sum_{u;\{x^l\};y} \alpha_1\Big(\{x^l\}, u, y\Big)\, r_{u,U_1}(y) q_{y,Y_1}\Big(\widehat{x}^L\Big) \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1}),
\end{aligned}
$$

where $\widehat{x}^0$ could also be denoted $\widehat{u}$, and, for all $t = 3, 4, ..., T+1$,

$$
\alpha_t(\{\widehat{x}^l\}) = \frac{q_{Y_{t-2},Y_{t-1}}\left(\widehat{x}^L\right)}{\overline{q}_{Y_{t-2},Y_{t-1}}} \sum_{\{x^l\}} \alpha_{t-1}\left(\{x^l\}\right) \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1}). \tag{3.4}
$$

Remark 3.1. *If not for initial distribution estimation, it would have been simpler to use the denominator $\overline{P}(U_{1,t-1}, Y_{1,t-1})$ instead of $\overline{P}(U_{2,t-1}, Y_{2,t-1}|U_1, Y_1)$ in the definition of α. Indeed, several small simplifications would occur with this natural replacement. However, the proper expectation-maximization-type update of initial distribution μ would then become problematic.*

Remark 3.2. *One issue in computing α on real problems when T is large is that α, essentially being the product of probability ratios, can vary dramatically and become extremely large or small. Hence, we will use the filter instead of α in our EM algorithm.*

To see how to recover the filter π from α, we first note by the definition of conditional expectation and

$$
\begin{aligned}
&\overline{E}\left[1_{U_{1,t}=u_{1,t},Y_{1,t}=y_{1,t}} \frac{P(\{X_t^l = x^l\}_{l=1}^L, U_{1,t}, Y_{1,t})}{\overline{P}(U_{1,t}, Y_{1,t})}\right] \\
&= \frac{P(\{X_t^l = x^l\}_{l=1}^L, U_{1,t} = u_{1,t}, Y_{1,t} = y_{1,t})}{\overline{P}(U_{1,t} = u_{1,t}, Y_{1,t} = y_{1,t})} \overline{E}\left(1_{U_{1,t}=u_{1,t},Y_{1,t}=y_{1,t}}\right) \\
&= P(\{X_t^l = x^l\}_{l=1}^L, U_{1,t} = u_{1,t}, Y_{1,t} = y_{1,t}) \\
&= \overline{E}\Big[1_{U_{1,t}=u_{1,t},Y_{1,t}=y_{1,t}} A_t 1_{\{X_t^l = x^l\}_{l=1}^L}\Big]
\end{aligned} \tag{3.5}
$$

that

$$
\begin{aligned}
\overline{E}\Big[A_t 1_{\{X_t^l\}=\{x^l\}} \mid U_{1,t}, Y_{1,t}\Big] &= \frac{P(U_{1,t}, Y_{1,t}, \{X_t^i = x^i\}_{i=1}^L)}{\overline{P}(U_{1,t}, Y_{1,t})} \\
&= \frac{P(U_{1,t}, Y_{1,t}, \{X_t^i = x^i\}_{i=1}^L)}{\overline{P}(U_{2,t}, Y_{2,t}\Big|U_1, Y_1)\overline{P}(U_1, Y_1)},
\end{aligned} \tag{3.6}
$$

where we just used the multiplication rule in the last equality, and summing over the $\{x^i\}$

$$
\overline{E}\left[A_t \mid U_{1,t}, Y_{1,t}\right] = \frac{P(U_{1,t}, Y_{1,t} \mid \theta)}{\overline{P}(U_{1,t}, Y_{1,t})} = \frac{P(U_{1,t}, Y_{1,t} \mid \theta)}{\overline{P}(U_{2,t}, Y_{2,t}\Big|U_1, Y_1)\overline{P}(U_1, Y_1)}. \tag{3.7}
$$

Then, by (3.1), Bayes' rule, (3.2), (3.6), and (3.7)

$$\begin{aligned} \pi_t(\{x^l\}) &= P\left(\{X_t^l = x^l\} \mid U_{1,t}, Y_{1,t}\right) \\ &= \frac{\overline{E}\left[A_t 1_{\{X_t^l\}=\{x^l\}} \mid U_{1,t}, Y_{1,t}\right]}{\overline{E}\left[A_t \mid U_{1,t}, Y_{1,t}\right]} \\ &= \frac{\alpha_{t+1}\left(\{x^l\}\right)}{b_t}, \quad \forall \{x^l\}, \end{aligned} \tag{3.8}$$

where b_t is the *forward Bayes' factor* b_t and full *Bayes' factor* B_t are defined as:

$$b_t = \frac{P\left(U_{1,t}, Y_{1,t} \mid \theta\right)}{\overline{P}\left(U_{2,t}, Y_{2,t} \mid U_1, Y_1\right)} = \sum_{\{x_t^l\}} \alpha_{t+1}(\{x_t^l\}), \quad \forall t \in \{1, 2, ..., T\} \tag{3.9}$$

and

$$\begin{aligned} B_t = \overline{E}\left[A_t \middle| U_{1,t}, Y_{1,t}\right] &= \frac{P\left(U_{1,t}, Y_{1,t} \mid \theta\right)}{\overline{P}\left(U_{1,t}, Y_{1,t}\right)} \\ &= \frac{b_t}{\overline{P}\left(U_1, Y_1\right)} \\ &= \frac{b_t}{\sum\limits_{u_0, y_0} \overline{\mu}_{UY}(u_0, y_0) r_{u_0, U_1}(y_0) \overline{q}_{y_0, Y_1}}, \quad \forall t \in \{1, 2, ..., T\}. \end{aligned} \tag{3.10}$$

In addition to being the filter, π is a normalized, shifted version of α that does not have extreme value problems that α does. As such, it is better to use a forward recursion for π than for α. To make this work effectively, we introduce a new variable ρ that is normalized like π except for the last step, where it is unnormalized like α.

Proposition 3.3. *The filter and forward Bayes' factor satisfy* $\pi_0 = \mu$, $b_0 = 1$

$$\rho_t(\{\widehat{x}^l\}) = \begin{cases} \sum\limits_{u, \{x^i\}, y} \pi_{t-1}\left(u, \{x^i\}, y\right) r_{u, U_1}(y) q_{y, Y_1}\left(\widehat{x}^L\right) \prod\limits_{l=1}^{L} p^l_{x^l, \widehat{x}^l}(\widehat{x}^{l-1}) & t = 1, \\ \frac{q_{Y_{t-1}, Y_t}(\widehat{x}^L)}{\overline{q}_{Y_{t-1}, Y_t}} \sum\limits_{\{x^i\}} \pi_{t-1}\left(\{x^i\}\right) \prod\limits_{l=1}^{L} p^l_{x^l, \widehat{x}^l}(\widehat{x}^{l-1}) & t > 1, \end{cases}$$

$$b_t = a_t b_{t-1}, \quad a_t = \sum_{\{x^l\}} \rho_t(\{x^l\}),$$

$$\pi_t(\{\widehat{x}^l\}) = \frac{\rho_t(\{\widehat{x}^l\})}{a_t},$$

for $t = 1, 2, ..., T$. *Then, the Bayes' factor is computed by*

$$B_t = \frac{b_t}{\sum\limits_{u_0, y_0} \overline{\mu}_{UY}(u_0, y_0) r_{u_0, U_1}(y_0) \overline{q}_{y_0, Y_1}}, \quad \forall t \in \{1, 2, ..., T\}. \tag{3.11}$$

Proof. The result follows from prior developments (3.4), (3.8)–(3.10) and the linearity of the equations. □

Due to cancellation, the Bayes' factor compares the target DBN model to the proposal model where inputs and outputs do not depend upon any hidden state (or input), but rather evolve as

$$(U_0, Y_0) \sim \overline{\mu}_{UY}, \quad \overline{P}\left(U_t = \widehat{u}, Y_t = \widehat{y} \middle| U_{t-1} = u, Y_{t-1} = y\right) = r_{u,\widehat{u}}(y)\overline{q}_{y,\widehat{y}}.$$

The one-step predictor and smoother follow from the filter as a simple corollary.

Corollary 3.4. *The one-step predictor satisfies:*

$$P\Big(\{X_{t+1}^l = x^l\} \,|\, U_{1,t}, Y_{1,t}\Big) = \sum_y q_{Y_t,y}\left(x^L\right) \sum_{\{x_t^l\}} \pi_t(\{x_t^l\}) \prod_{l=1}^{L} p^l_{x_t^l, x^l}(x^{l-1}), \ \forall \{x^l\} \tag{3.12}$$

for $t = 1, 2, ..., T$ while the one-step smoother satisfies:

$$P\Big(\{X_{t-1}^l = x^l\} \,|\, U_{1,t}, Y_{1,t}\Big) = \begin{cases} \frac{b_{t-1}}{b_t} \sum\limits_{\{x_t^l\}} \frac{q_{Y_{t-1},Y_t}(x_t^L)}{\overline{q}_{Y_{t-1},Y_t}} \pi_{t-1}(\{x^l\}) \prod\limits_{l=1}^{L} p^l_{x^l, x_t^l}(x_t^{l-1}) & t = 2, ..., T, \\ \frac{1}{b_1} \sum\limits_{u,\{x_1^l\},y} q_{y,Y_1}\left(x_1^L\right) r_{u,U_1}(y) \pi_0(\{x^l\}, u, y) \prod\limits_{l=1}^{L} p^l_{x^l, x_1^l}(x_1^{l-1}) & t = 1. \end{cases} \tag{3.13}$$

Proof. We first consider the one step predictor. First, it follows by (3.2), (3.6) and the tower property that

$$\overline{E}\Big[A_{t+1} 1_{\{X_{t+1}^l = \{x^l\}}} \,|\, U_{1,t}, Y_{1,t}\Big] = \frac{\overline{E}\Big[\alpha_{t+2}\Big(\{x^l\}\Big) \,|\, U_{1,t}, Y_{1,t}\Big]}{\overline{P}(U_1, Y_1)}. \tag{3.14}$$

Hence, by Bayes' rule, the martingale property, the tower property, (3.4), (3.8) and (3.10)

$$\begin{aligned} P\Big(\{X_{t+1}^l\} = \{x^l\} \,|\, U_{1,t}, Y_{1,t}\Big) &= \frac{\overline{E}\Big[A_{t+1} 1_{\{X_{t+1}^l = \{x^l\}}} \,|\, U_{1,t}, Y_{1,t}\Big]}{\overline{E}[A_t \mid U_{1,t}, Y_{1,t}]} \\ &= \frac{\overline{E}\Big[\alpha_{t+2}\Big(\{x^l\}\Big) \,|\, U_{1,t}, Y_{1,t}\Big]}{\overline{P}(U_1, Y_1) B_t} \\ &= \frac{\overline{E}\left[\frac{q_{Y_t,Y_{t+1}}(x^L)}{\overline{q}_{Y_t,Y_{t+1}}} \sum\limits_{\{x_t^l\}} \alpha_{t+1}\Big(\{x_t^l\}\Big) \prod\limits_{l=1}^{L} p^l_{x_t^l, x^l}(x^{l-1}) \,|\, U_{1,t}, Y_{1,t}\right]}{b_t} \\ &= \overline{E}\left[\frac{q_{Y_t,Y_{t+1}}(x^L)}{\overline{q}_{Y_t,Y_{t+1}}} \,|\, Y_t\right] \sum_{\{x_t^l\}} \pi_t(\{x_t^l\}) \prod_{l=1}^{L} p^l_{x_t^l, x^l}(x^{l-1}) \\ &= \sum_y q_{Y_t,y}\left(x^L\right) \sum_{\{x_t^l\}} \pi_t(\{x_t^l\}) \prod_{l=1}^{L} p^l_{x_t^l, x^l}(x^{l-1}), \ \forall \{x^l\} \end{aligned} \tag{3.15}$$

for $t = 1, 2, ..., T$.

Turning to the smoother, one finds from Bayes' rule, (2.1) and (3.10) as well as independence under the proposal model that for $t = 2, ..., T$

$$
\begin{aligned}
P\Big(\{X_{t-1}^l = x^l\}\,|\,U_{1,t}, Y_{1,t}\Big) &= \frac{\overline{E}\left[A_t 1_{\{X_{t-1}^l = x^l\}_{l=1}^L}\,|\,U_{1,t}, Y_{1,t}\right]}{B_t} \qquad (3.16)\\
&= \frac{\overline{E}\left[\frac{q_{Y_{t-1},Y_t}(X_t^L)}{\overline{q}_{Y_{t-1},Y_t}} A_{t-1} \prod_{l=1}^L \frac{p_{x^l,x_t^l}(X_t^{l-1})}{\overline{p}_{x^l,x_t^l}} 1_{\{X_{t-1}^l = x^l\}}\,|\,U_{1,t}, Y_{1,t}\right]}{B_t}\\
&= \frac{\sum_{\{x_t^l\}} \frac{q_{Y_{t-1},Y_t}(x_t^L)}{\overline{q}_{Y_{t-1},Y_t}} \overline{E}\left[A_{t-1} 1_{\{X_{t-1}^l = x^l\}} | U_{1,t}, Y_{1,t}\right] \prod_{l=1}^L p_{x^l,x_t^l}^l(x_t^{l-1})}{B_t}\\
&= \sum_{\{x_t^l\}} \frac{q_{Y_{t-1},Y_t}\left(x_t^L\right)}{\overline{q}_{Y_{t-1},Y_t}} \frac{\alpha_t(\{x^l\})}{b_t} \prod_{l=1}^L p_{x^l,x_t^l}^l(x_t^{l-1})\\
&= \frac{b_{t-1}}{b_t} \sum_{\{x_t^l\}} \frac{q_{Y_{t-1},Y_t}\left(x_t^L\right)}{\overline{q}_{Y_{t-1},Y_t}} \pi_{t-1}(\{x^l\}) \prod_{l=1}^L p_{x^l,x_t^l}^l(x_t^{l-1}),
\end{aligned}
$$

where we used independence, (3.2) and (3.6) again in the second last equality. It follows from Bayes' rule and the Markov property for (U, Y) under $\overline{P}$

$$
\begin{aligned}
\overline{P}\left(U_0 = u, Y_0 = y \middle| U_1, Y_1\right) &= \frac{r_{u,U_1}(y)\overline{q}_{y,Y_1}\overline{P}(U_0 = u, Y_0 = y)}{\overline{P}(U_1, Y_1)} \qquad (3.17)\\
&= \frac{r_{u,U_1}(y)\overline{q}_{y,Y_1}\overline{\mu}_{UY}(u, y)}{\sum_{u',y'} r_{u',U_1}(y')\overline{q}_{y',Y_1}\overline{\mu}_{UY}(u', y')},
\end{aligned}
$$

so when $t = 1$

$$
\begin{aligned}
P\Big(\{X_0^l = x^l\}\,|\,U_1, Y_1\Big) &= \frac{\overline{E}\left[A_1 1_{\{X_0^l = x^l\}_{l=1}^L}\,|\,U_1, Y_1\right]}{B_1} \qquad (3.18)\\
&= \frac{\overline{E}\left[\frac{\mu(U_0,x^1,\ldots,x^L,Y_0)}{\overline{\mu}_1(x^1)\cdots\overline{\mu}_L(x^L)\overline{\mu}_{UY}(U_0,Y_0)} \frac{q_{Y_0,Y_1}(X_1^L)}{\overline{q}_{Y_0,Y_1}} \prod_{l=1}^L \frac{p_{x^l,X_1^l}(X_1^{l-1})}{\overline{p}_{x^l,X_1^l}} 1_{\{X_0^l = x^l\}}\,|\,U_1, Y_1\right]}{B_1}\\
&= \frac{\sum_{u,\{x_1^l\},y} \frac{q_{y,Y_1}(x_1^L) r_{u,U_1}(y)}{\sum_{u',y'} r_{u',U_1}(y')\overline{q}_{y',Y_1}\overline{\mu}_{UY}(u',y')} \mu(u, x^1, \ldots, x^L, y) \prod_{l=1}^L p_{x^l,x_1^l}^l(x_1^{l-1})}{B_1}\\
&= \frac{1}{b_1} \sum_{u,\{x_1^l\},y} q_{y,Y_1}\left(x_1^L\right) r_{u,U_1}(y)\pi_0(\{x^l\}_{l=1}^L, u, y) \prod_{l=1}^L p_{x^l,x_1^l}^l(x_1^{l-1}),
\end{aligned}
$$

since $b_1 = \overline{P}(U_1, Y_1)B_1$. □

4. Path ratios and path filters

The filter, as defined above, estimates the current hidden state based upon the historical back observations of the inputs and outputs. However, in many cases such as batch learning one has access

to whole sequence observations from which to make estimates. In this case, we can define the *path filter*

$$\begin{array}{rll} \Phi_t(\{x^l\}) &= P(\{X_t^l = x^l\} \mid U_{1,T}, Y_{1,T}) & t \in \{1, 2, ..., T\}, \\ \Phi_0(\{x^l\}, u, y) &= P(U_0 = u, \{X_0^l = x^l\}, Y_0 = y \mid U_{1,T}, Y_{1,T}) & t = 0, \end{array} \tag{4.1}$$

and

$$\begin{array}{rll} \Phi_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) = P(\{X_{t-1}^l = x^l\}, \{X_t^l = \widehat{x}^l\} \mid U_{1,T}, Y_{1,T}) & & t \geq 2, \\ \Phi_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) = P(U_0 = u, \{X_0^l = x^l\}, Y_0 = y, \{X_1^l = \widehat{x}^l\} \mid U_{1,T}, Y_{1,T}) & & t = 1, \end{array} \tag{4.2}$$

that estimates the hidden states at all levels at the current and prior time given all observations. Then, by (4.2), Bayes' rule and the argument in (3.6) and (3.7)

$$\begin{aligned} \Phi_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) &= \frac{\overline{E}\left[A_T 1_{\{X_{t-1}^l = x^l\}, \{X_t^l = \widehat{x}^l\}} \mid U_{1,T}, Y_{1,T}\right]}{\overline{E}\left[A_T \mid U_{1,T}, Y_{1,T}\right]} \\ &= \frac{\nu_{t-1,t}\left(\{x^l\}, \{\widehat{x}^l\}\right)}{b_T}, \quad \forall \{x^l\}, \{\widehat{x}^l\}, \end{aligned} \tag{4.3}$$

for $t = 2, 3, ..., T$, where $b_T = \overline{E}\left[A_T \mid U_{1,T}, Y_{1,T}\right] \overline{P}(U_1, Y_1)$ is the forward Bayes' factor defined in (3.9) and full path ratio is defined as

$$\begin{aligned} \nu_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) &= \frac{P(U_{1,T}, Y_{1,T}, \{X_{t-1}^l = x^l\}, \{X_t^l = \widehat{x}^l\})}{\overline{P}(U_{2,T}, Y_{2,T} \mid U_1, Y_1)} \\ &= \overline{E}\left[A_T 1_{\{X_{t-1}^l = x^l\}, \{X_t^l = \widehat{x}^l\})} \mid U_{1,T}, Y_{1,T}\right] \overline{P}(U_1, Y_1), \end{aligned} \tag{4.4}$$

for $t = 2, 3, ..., T$. Here we used the definition of conditional expectation again. Similarly, in the case $t = 1$ one finds

$$\Phi_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) = \frac{\nu_{0,1}\left(\{x^l\}, u, y, \{\widehat{x}^l\}\right)}{b_T}, \quad \forall \{x^l\}, \{\widehat{x}^l\}, \tag{4.5}$$

and full path ratio is defined as

$$\begin{aligned} \nu_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) &= \frac{P(U_{1,T}, Y_{1,T}, U_0 = u, \{X_0^l = x^l\}, Y_0 = y, \{X_1^l = \widehat{x}^l\})}{\overline{P}(U_{2,T}, Y_{2,T} \mid U_1, Y_1)} \\ &= \overline{E}\left[A_T 1_{U_0 = u, \{X_0^l = x^l\}, Y_0 = y, \{X_1^l = \widehat{x}^l\})} \mid U_{1,T}, Y_{1,T}\right] \overline{P}(U_1, Y_1). \end{aligned} \tag{4.6}$$

The forward ratio α for filter π has a nice forward recursion but is only based upon the back observations. We often want estimates based upon the whole collection of observations, meaning we want to use Φ. The path ratio ν for Φ does not have a recursion. However, α or π can be combined with a backward variable to form the path filter Φ as follows: First, the backward ratio is defined as

$$\beta_t(\{\widehat{x}^l\}) = \frac{P(U_{t+1,T}, Y_{t+1,T} \mid U_t, Y_t, \{X_t^l = \widehat{x}^l\}, \theta)}{\overline{P}(U_{t+1,T}, Y_{t+1,T} \mid U_t, Y_t)} \tag{4.7}$$

for $t = 1, ..., T-1, T$ (so $\beta_T = 1$) with

$$\beta_0(\{\widehat{x}^l\}, u, y) = \frac{P(U_{1,T}, Y_{1,T} \mid \{X_0^l = \widehat{x}^l\}, U_0 = u, Y_0 = y, \theta)}{\overline{P}(U_{1,T}, Y_{1,T} \mid U_0 = u, Y_0 = y)}. \tag{4.8}$$

Then, the path ratio is reduced to the forward and backward ratios by the definition of conditional expectation, the multiplication rule and (3.2), (4.4), (4.7)

$$\begin{aligned}
&\nu_{t-1,t}(\{x^l\},\{\widehat{x}^l\}) \\
=&\frac{P(\{X_{t-1}^l=x^l\},\{X_t^l=\widehat{x}^l\},U_{1,T},Y_{1,T})}{\overline{P}(U_{2,T},Y_{2,T}\mid U_1,Y_1)} \\
=&\frac{P(\{X_{t-1}^l=x^l\},U_{1,t-1},Y_{1,t-1})r_{U_{t-1},U_t}(Y_{t-1})P(U_{t+1,T},Y_{t+1,T}\big|\{X_t^l=\widehat{x}^l\},U_t,Y_t)}{\overline{P}(U_{2,t-1},Y_{2,t-1}\big|U_1,Y_1)r_{U_{t-1},U_t}(Y_{t-1})\overline{q}_{Y_{t-1},Y_t}\overline{P}(U_{t+1,T},Y_{t+1,T}\mid U_t,Y_t)} \\
&\times q_{Y_{t-1},Y_t}(\widehat{x}^L)\left[\prod_{i=1}^{L}p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right] \\
=&\alpha_t\left(\{x^l\}\right)\frac{\left[\prod_{i=1}^{L}p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right]q_{Y_{t-1},Y_t}(\widehat{x}^L)}{\overline{q}_{Y_{t-1},Y_t}}\beta_t\left(\{\widehat{x}^l\}\right),
\end{aligned} \tag{4.9}$$

for all $t=2,3,...,T$. Hence, the path filter decomposition follows from (3.8), (4.3), (4.9), and Proposition 3.3:

$$\begin{aligned}
\Phi_{t-1,t}(\{x^l\},\{\widehat{x}^l\})&=\frac{\nu_{t-1,t}(\{x^l\},\{\widehat{x}^l\})}{b_T} \\
&=\pi_{t-1}\left(\{x^l\}\right)\frac{\left[\prod_{i=1}^{L}p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right]q_{Y_{t-1},Y_t}(\widehat{x}^L)}{\overline{q}_{Y_{t-1},Y_t}}\chi_t\left(\{\widehat{x}^l\}\right),
\end{aligned} \tag{4.10}$$

for all $t=2,3,...,T$, where

$$\chi_t\left(\{\widehat{x}^l\}\right)=\frac{b_{t-1}}{b_T}\beta_t\left(\{\widehat{x}^l\}\right)=\frac{\beta_t\left(\{\widehat{x}^l\}\right)}{a_t\cdots a_T},\ \ \forall\{\widehat{x}^l\},\ t=1,2,...,T. \tag{4.11}$$

Similarly, in the case $t=1$ one finds that

$$\Phi_{0,1}(\{x^l\},u,y,\{\widehat{x}^l\})=\pi_0\left(u,\{x^l\},y\right)\frac{\left[\prod_{i=1}^{L}p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right]q_{y,Y_1}(\widehat{x}^L)}{\overline{q}_{y,Y_1}}\chi_1\left(\{\widehat{x}^l\}\right). \tag{4.12}$$

Now, it follows by (4.7), (4.8) as well as the Markov properties with P and $\overline{P}$ that

$$\begin{aligned}
&\beta_t\left(\{\widehat{x}^l\}\right) \\
=&\sum_{\{x^l\}}\frac{P(U_{t+1},Y_{t+1},\{X_{t+1}^l=x^l\}\big|U_t,Y_t,\{X_t^l=\widehat{x}^l\})P(U_{t+2,T},Y_{t+2,T}\big|U_{t+1},Y_{t+1},\{X_{t+1}^l=x^l\})}{\overline{P}(U_{t+1},Y_{t+1}\mid U_t,Y_t)\overline{P}(U_{t+2,T},Y_{t+2,T}\big|U_{t+1},Y_{t+1})} \\
=&\sum_{\{x^l\}}\frac{q_{Y_t,Y_{t+1}}\left(x^L\right)}{\overline{q}_{Y_t,Y_{t+1}}}\beta_{t+1}\left(\{x^l\}\right)\prod_{i=1}^{L}p^i_{\widehat{x}^i,x^i}(x^{i-1})
\end{aligned} \tag{4.13}$$

for all $t=T-1,T-2,...,1$ starting from $\beta_T=1$ and then

$$\beta_0\left(\{\widehat{x}^l\},u,y\right)=\beta_0\left(\{\widehat{x}^l\},y\right)=\sum_{\{x^l\}}\frac{q_{y,Y_1}\left(x^L\right)}{\overline{q}_{y,Y_1}}\beta_1\left(\{x^l\}\right)\prod_{i=1}^{L}p^i_{\widehat{x}^i,x^i}(x^{i-1}) \tag{4.14}$$

(does not actually depend upon u). Thus,

$$\chi_t\left(\{\widehat{x}^l\}\right) = \sum_{\{x^l\}} \frac{q_{Y_t,Y_{t+1}}\left(x^L\right)}{a_t\ \overline{q}_{Y_t,Y_{t+1}}} \chi_{t+1}\left(\{x^l\}\right) \prod_{i=1}^{L} p^i_{\widehat{x}^i,x^i}(x^{i-1}) \tag{4.15}$$

for all $t = T-1, T-2, ..., 1$ starting from $\chi_T = \frac{1}{a_T}$ and then

$$\chi_0\left(\{\widehat{x}^l\}, y\right) = \sum_{\{x^l\}} \frac{q_{y,Y_1}\left(x^L\right)}{\overline{q}_{y,Y_1}} \chi_1\left(\{x^l\}\right) \prod_{i=1}^{L} p^i_{\widehat{x}^i,x^i}(x^{i-1}). \tag{4.16}$$

We can now state the efficient method to compute the path filter:

Theorem 4.1. *Suppose that filters π_t and ρ_t are as in Proposition 3.3. Let backward recursion for χ be given by (4.15), (4.16) starting with $\chi_T = \frac{1}{a_T}$. Then, the path filter satisfies:*

$$\Phi_t(\{\widehat{x}^l\}) = \rho_t\Big(\{x^l\}\Big)\chi_t\left(\{x^l\}\right),\ \forall t = 1, ..., T, \tag{4.17}$$

$$\Phi_0(u, \{\widehat{x}^l\}, y) = \pi_0\Big(u, \{x^l\}, y\Big)\chi_0\left(\{x^l\}, y\right), \tag{4.18}$$

$$\Phi_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) = \pi_{t-1}\Big(\{x^l\}\Big)\frac{\left[\prod_{i=1}^{L} p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right]q_{Y_{t-1},Y_t}(\widehat{x}^L)}{\overline{q}_{Y_{t-1},Y_t}}\chi_t\Big(\{\widehat{x}^l\}\Big),\ \forall t = 2, ..., T, \tag{4.19}$$

$$\Phi_{0,1}(\{x^l\},u, y, \{\widehat{x}^l\}) = \pi_0\Big(u, \{x^l\}, y\Big)\frac{\left[\prod_{i=1}^{L} p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right]q_{y,Y_1}(\widehat{x}^L)}{\overline{q}_{y,Y_1}}\chi_1\Big(\{\widehat{x}^l\}\Big) \tag{4.20}$$

for all $u, y, \{x^l\}, \{\widehat{x}^l\}$.

Proof. The final two equations are established above. The prior ones are then established by summing out the appropriate index and substitution. □

5. DBN expectation maximization algorithm

In this section, we consider learning all the parameters of the DBN model. Since the input rate $r_{u,\widehat{u}}(y)$ only depends upon inputs and outputs, all of which are observable during the training period, this rate can be learned by elementary methods and we will move to the other parameters.

Turning to the estimate for the initial condition, we find from Bayes' rule, total probability and the DBN assumptions that

$$\begin{aligned}\breve{\mu}(u, \{x^l\}, y) =& P\left(U_0 = u, \{X^l_0 = x^l\}, Y_0 = y \middle| U_{1,T}, Y_{1,T}\right) \\ =& \frac{P\left(U_{1,T}, Y_{1,T}\middle|\{X^l_0 = x^l\}, U_0 = u, Y_0 = y\right)P\left(\{X^l_0 = x^l\}, U_0 = u, Y_0 = y\right)}{P\left(U_{1,T}, Y_{1,T}\right)}.\end{aligned} \tag{5.1}$$

Hence, if we divide numerator and denominator of (5.1) by

$$\overline{P}(U_{1,T}, Y_{1,T}|U_0 = u, Y_0 = y) = r_{u,U_1}(y)\overline{q}_{y,Y_1}\left[\prod_{j=2}^{T} r_{U_{j-1},U_j}(Y_{j-1})\overline{q}_{Y_{j-1},Y_j}\right], \tag{5.2}$$

then we get from (4.8) and (3.9) the initial distribution update

$$\begin{aligned}\breve{\mu}\left(u,\{x^l\},y\right) &= \frac{r_{u,U_1}(y)\overline{q}_{y,Y_1}\beta_0(\{x^l\},u,y)\mu\left(u,\{x^l\},y\right)}{b_T} \\ &= r_{u,U_1}(y)\overline{q}_{y,Y_1}\chi_0(\{x^l\},u,y)\mu\left(u,\{x^l\},y\right),\end{aligned} \tag{5.3}$$

which is already reduced to the forward and backward ratios as well as the prior estimates.

Assigning the new rate to be the expected average rate, we have that:

$$\begin{aligned}\breve{p}^1_{x^1,\widehat{x}^1}(u) &= \frac{\sum\limits_{t=1}^{T} P\left(X^1_{t-1}=x^1, X^1_t=\widehat{x}^1, U_t=u\middle|U_{1,T},Y_{1,T}\right)}{\sum\limits_{t=1}^{T} P\left(X^1_{t-1}=x^1, U_t=u\middle|U_{1,T},Y_{1,T}\right)} \\ &= \frac{\sum\limits_{\widehat{u},y}\sum\limits_{\widehat{x}^{2,L}}\sum\limits_{x^{2,L}}\Phi_{0,1}(\{x^l\},\widehat{u},y,\{\widehat{x}^l\})1_{U_1=u}+\sum\limits_{t=2}^{T}\sum\limits_{\widehat{x}^{2,L}}\sum\limits_{x^{2,L}}\Phi_{t-1,t}(\{x^l\},\{\widehat{x}^l\})1_{U_t=u}}{\sum\limits_{\widehat{u},y}\sum\limits_{x^{2,L}}\Phi_0(\widehat{u},\{x^l\},y)1_{U_1=u}+\sum\limits_{t=2}^{T}\sum\limits_{x^{2,L}}\Phi_{t-1}(\{x^l\})1_{U_t=u}}.\end{aligned} \tag{5.4}$$

Similarly, the new estimates for the rates for the i^{th} hidden layer are:

$$\begin{aligned}\breve{p}^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1}) &= \frac{\sum\limits_{t=1}^{T} P\left(X^i_{t-1}=x^i, X^i_t=\widehat{x}^i, X^{i-1}_t=\widehat{x}^{i-1}\middle|U_{1,T},Y_{1,T}\right)}{\sum\limits_{t=1}^{T} P\left(X^i_{t-1}=x^i, X^{i-1}_t=\widehat{x}^{i-1}\middle|U_{1,T},Y_{1,T}\right)} \\ &= \frac{\sum\limits_{\{\widehat{\xi}^l\}}\sum\limits_{\{\xi^l\}}\left[\sum\limits_{u,y}\Phi_{0,1}(\{\xi^l\},u,y,\{\widehat{\xi}^l\})+\sum\limits_{t=2}^{T}\Phi_{t-1,t}(\{\xi^l\},\{\widehat{\xi}^l\})\right]1_{\xi^i=x^i,\widehat{\xi}^{i-1}=\widehat{x}^{i-1},\widehat{\xi}^i=\widehat{x}^i}}{\sum\limits_{\{\widehat{\xi}^l\}}\sum\limits_{\{\xi^l\}}\left[\sum\limits_{u,y}\Phi_{0,1}(\{\xi^l\},u,y,\{\widehat{\xi}^l\})+\sum\limits_{t=2}^{T}\Phi_{t-1,t}(\{\xi^l\},\{\widehat{\xi}^l\})\right]1_{\xi^i=x^i,\widehat{\xi}^{i-1}=\widehat{x}^{i-1}}}\end{aligned} \tag{5.5}$$

for $i \in \{2,...,L\}$ and the new output layer rate estimates are:

$$\breve{q}_{y,\widehat{y}}(\widehat{x_L}) = \frac{\sum\limits_{u}\sum\limits_{\{x^l\}}\sum\limits_{\widehat{x}^{1,L-1}}\Phi_{0,1}(\{x^l\},u,y,\{\widehat{x}^l\})1_{Y_1=\widehat{y}}+\sum\limits_{t=2}^{T}\sum\limits_{\widehat{x}^{1,L-1}}\Phi_t(\{\widehat{x}^l\})1_{Y_{t-1}=y,Y_t=\widehat{y}}}{\sum\limits_{u}\sum\limits_{\{x^l\}}\sum\limits_{\widehat{x}^{1,L-1}}\Phi_{0,1}(\{x^l\},u,y,\{\widehat{x}^l\})+\sum\limits_{t=2}^{T}\sum\limits_{\widehat{x}^{1,L-1}}\Phi_t(\{\widehat{x}^l\})1_{Y_{t-1}=y}}. \tag{5.6}$$

Now, we have all the equations required for the forward-backward expectation-maximization algorithm to determine the system parameters, which is given in Algorithm 1.

Algorithm 1: EM algorithm for DBN.

Data: Input, Output sequence: $U_1, Y_1; U_2, Y_2; ...; U_T, Y_T$
Input: Initial Estimates: $\{p^l_{x^l \to \widehat{x}^l}(\widehat{x}^{l-1})\}, q_{y\to\widehat{y}}(\widehat{x}^L), \mu(u, \{x^l\}, y)$
Output: Final Estimates: $\{p^l_{x^l \to \widehat{x}^l}(\widehat{x}^{l-1})\}, q_{y\to\widehat{y}}(\widehat{x}^L), \mu(u, \{x^l\}, y)$
Initialization: $\pi_0 = \mu$; $b_0 = 1$

while *p, q, and* μ *have not converged* **do**

/* Compute forward Bayes' and Filter. */

$$\rho_1(\{\widehat{x}^i\}) = \sum_{u,\{x^i\},y} \pi_0\left(u, \{x^i\}, y\right) r_{u,U_1}(y) q_{y,Y_1}\left(\widehat{x}^L\right) \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1})$$

for $t = 1, 2, ..., T$ **do**

$$a_t = \sum_{\{x^l\}} \rho_t(\{x^l\}), \quad b_t = a_t b_{t-1}, \quad \pi_t(\{\widehat{x}^l\}) = \frac{\rho_t(\{\widehat{x}^l\})}{a_t}.$$

If $t \neq T$ then $\rho_{t+1}(\{\widehat{x}^i\}) = \frac{q_{Y_t,Y_{t+1}}(\widehat{x}^L)}{\overline{q}_{Y_t,Y_{t+1}}} \sum_{\{x^l\}} \pi_t\left(\{x^i\}\right) \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1})$.

/* Backward propagation and Path Filter. */

$$\chi_T = \frac{1}{a_T}, \Phi_T(\{x^l\}) = \rho_T\left(\{x^l\}\right)\chi_T$$

for $t = T-1, T-2, ..., 1$ **do**

$$\chi_t\left(\{\widehat{x}^l\}\right) = \sum_{\{x^l\}} \frac{q_{Y_t,Y_{t+1}}(x^L)}{a_t\, \overline{q}_{Y_t,Y_{t+1}}} \chi_{t+1}\left(\{x^l\}\right) \prod_{i=1}^{L} p^i_{\widehat{x}^i,x^i}(x^{i-1})$$

$$\Phi_t(\{x^l\}) = \rho_t\left(\{x^l\}\right)\chi_t\left(\{x^l\}\right)$$

$$\chi_0\left(\{\widehat{x}^l\}, u, y\right) = \sum_{\{x^l\}} \frac{q_{y,Y_1}(x^L)}{\overline{q}_{y,Y_1}} \chi_1\left(\{x^l\}\right) \prod_{i=1}^{L} p^i_{\widehat{x}^i,x^i}(x^{i-1})$$

$$\Phi_0(u, \{x^l\}, y) = \pi_0\left(u, \{x^l\}, y\right)\chi_0\left(\{x^l\}, u, y\right)$$

for $t = 2, ..., T$ **do**

$$\Phi_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) = \pi_{t-1}\left(\{x^l\}\right) \frac{\left[\prod_{i=1}^{L} p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right] q_{Y_{t-1},Y_t}(\widehat{x}^L)}{\overline{q}_{Y_{t-1},Y_t}} \chi_t\left(\{\widehat{x}^l\}\right)$$

$$\Phi_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) = \pi_0\left(u, \{x^l\}, y\right) \frac{\left[\prod_{i=1}^{L} p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right] q_{y,Y_1}(\widehat{x}^L)}{\overline{q}_{y,Y_1}} \chi_1\left(\{\widehat{x}^l\}\right)$$

/* Probability Updates. */

$$\breve{\mu}\left(u, \{x^l\}, y\right) = r_{u,U_1}(y)\overline{q}_{y,Y_1}\chi_0(\{x^l\}, u, y)\mu\left(u, \{x^l\}, y\right).$$

for $i = 2, ..., L$ **do**

$$\breve{p}^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1}) = \frac{\sum_{\{\widehat{\xi}^l\}} \sum_{\{\xi^l\}} \left[\sum_{u,y} \Phi_{0,1}(\{\xi^l\}, u, y, \{\widehat{\xi}^l\}) + \sum_{t=2}^{T} \Phi_{t-1,t}(\{\xi^l\}, \{\widehat{\xi}^l\})\right] 1_{\xi^i = x^i, \widehat{\xi}^{i-1} = \widehat{x}^{i-1}, \widehat{\xi}^i = \widehat{x}^i}}{\sum_{\{\widehat{\xi}^l\}} \sum_{\{\xi^l\}} \left[\sum_{u,y} \Phi_{0,1}(\{\xi^l\}, u, y, \{\widehat{\xi}^l\}) + \sum_{t=2}^{T} \Phi_{t-1,t}(\{\xi^l\}, \{\widehat{\xi}^l\})\right] 1_{\xi^i = x^i, \widehat{\xi}^{i-1} = \widehat{x}^{i-1}}}$$

$$\breve{p}^1_{x^1,\widehat{x}^1}(u) = \frac{\sum_{\widehat{u},y} \sum_{\widehat{x}^{2,L}} \sum_{x^{2,L}} \Phi_{0,1}(\{x^l\}, \widehat{u}, y, \{\widehat{x}^l\}) 1_{U_1 = u} + \sum_{t=2}^{T} \sum_{\widehat{x}^{2,L}} \sum_{x^{2,L}} \Phi_{t-1,t}(\{x^l\}, \{\widehat{x}^l\}) 1_{U_t = u}}{\sum_{\widehat{u},y} \sum_{x^{2,L}} \Phi_0(\widehat{u}, \{x^l\}, y) 1_{U_1 = u} + \sum_{t=2}^{T} \sum_{x^{2,L}} \Phi_{t-1}(\{x^l\}) 1_{U_t = u}},$$

$$\breve{q}_{y,\widehat{y}}(\widehat{x_L}) = \frac{\sum_{u} \sum_{\{x^l\}} \sum_{\widehat{x}^{1,L-1}} \Phi_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) 1_{Y_1 = \widehat{y}} + \sum_{t=2}^{T} \sum_{\widehat{x}^{1,L-1}} \Phi_t(\{\widehat{x}^l\}) 1_{Y_{t-1} = y, Y_t = \widehat{y}}}{\sum_{u} \sum_{\{x^l\}} \sum_{\widehat{x}^{1,L-1}} \Phi_{0,1}(\{x^l\}, u, y, \{\widehat{x}^l\}) + \sum_{t=2}^{T} \sum_{\widehat{x}^{1,L-1}} \Phi_t(\{\widehat{x}^l\}) 1_{Y_{t-1} = y}}.$$

Note: The following three potential speed up tricks were not included in the algorithm for assimilation purposes. However, in some cases they prove quite useful.

(1) One can compute $t \to p(x^1, ..., x^L, \widehat{x}^1, ..., \widehat{x}^L, U_{t+1}, Y_t, Y_{t+1}) = \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1}) q_{Y_t,Y_{t+1}}(\widehat{X}^L)$ once before the computation of the unnormalized filter ρ and then reuse thereafter.

(2) If there is no natural canonical, stateless model, then we can choose $\overline{q} = \frac{1}{M}$, where M is the number of observation states, and then remove $\overline{q}$ from the calculations e.g.,

$$\rho_t(\{\widehat{x}^i\}) = q_{Y_{t-1},Y_t}\left(\widehat{x}^L\right) \sum_{\{x^i\}} \pi_{t-1}\left(\{x^i\}\right) \prod_{l=1}^{L} p^l_{x^l,\widehat{x}^l}(\widehat{x}^{l-1}).$$

$$\chi_t\left(\{\widehat{x}^l\}\right) = \sum_{\{x^l\}} \frac{q_{Y_t,Y_{t+1}}\left(x^L\right)}{a_t} \chi_{t+1}\left(\{x^l\}\right) \prod_{i=1}^{L} p^i_{\widehat{x}^i,x^i}(x^{i-1}).$$

$$\Phi_{t-1,t} = M\pi_{t-1}\left(\{x^l\}\right)\left[\prod_{i=1}^{L} p^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})\right] q_{Y_{t-1},Y_t}(\widehat{x}^L)\chi_t\left(\{\widehat{x}^l\}\right).$$

(3) First compute the numerator of the probabilities like

$$\sum_{\{\widehat{\xi}^l\}} \sum_{\{\xi^l\}} \left[\sum_{u,y} \Phi_{0,1}(\{\xi^l\}, u, y, \{\widehat{\xi}^l\}) + \sum_{t=2}^{T} \Phi_{t-1,t}(\{\xi^l\}, \{\widehat{\xi}^l\})\right] 1_{\xi^i = x^i, \widehat{\xi}^{i-1} = \widehat{x}^{i-1}, \widehat{\xi}^i = \widehat{x}^i}$$

for $\check{p}^i_{x^i,\widehat{x}^i}(\widehat{x}^{i-1})$ and then normalize that to form a probability mass function.

6. Conclusions

A probabilistic deep recurrent neural network, termed the deep Bayesian Network (DBN), which is parameterized in a different manner was introduced. This DBN generalizes the Markov Observation Model and the Pairwise Markov Chain models developed in [9] and thereby also generalizes the hidden Markov model. Each layer, including the input and output, is a Markov chain that depends the previous layer. Herein DBN's powerful filtering, learning, simulation and Bayes' factor capabilities were established mathematically. These can be use for example to provide an effective generative adversarial network (GAN) in the sequential (RNN) setting.

The proofs herein developed the method based upon the novel idea to transform the whole network to a completely independent network where the analysis is trivial using a Girsanov like theorem. This idea in its own right may turn out to be important.

The given DBN algorithm has been tested by students on up to three hidden levels on very simple problems. However, it is believed that the given algorithm can be made far more computer efficient and that DBNs may be better suited GPU machines and serious problems. We look forward to seeing where this technology might go.

Use of Generative-AI tools declaration

The author declares he/she has not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

Prof. Michael A. Kouritzin is the Guest Editor of special issue "The Mathematics of Artificial Intelligence" for AIMS Mathematics. Prof. Michael A. Kouritzin was not involved in the editorial review and the decision to publish this article.

The author declares no conflict of interest in this paper.

References

1. S. I. Amari, Characteristics of random nets of analog neuron-like elements, *IEEE Trans. Syst., Man, Cybern.*, **SMC-2** (1972), 643–657. https://doi.org/10.1109/TSMC.1972.4309193
2. N. Chopin, O. Papaspiliopoulos, *An introduction to sequential Monte Carlo*, Springer Cham, 2020. https://doi.org/10.1007/978-3-030-47845-2
3. D. Creal, A survey of sequential Monte Carlo methods for economics and finance, *Economet. Rev.*, **31** (2012), 245–296. https://doi.org/10.1080/07474938.2011.607333
4. E. D'Amato, V. A. Nardi, I. Notaro, V. Scordamaglia, A particle filtering approach for fault detection and isolation of UAV IMU sensors: design, implementation and sensitivity analysis, *Sensors*, **21** (2021), 3066. https://doi.org/10.3390/s21093066
5. J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, **79** (1982), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554
6. S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, **06** (1998), 107–116. https://doi.org/10.1142/S0218488598000094
7. E. Ising, Beitrag zur theorie des ferromagnetismus, *Z. Phys.*, **31** (1925), 253–258. https://doi.org/10.1007/BF02980577
8. M. A. Kouritzin, Sampling and filtering with Markov chains, *Signal Process.*, **225** (2024), 109613. https://doi.org/10.1016/j.sigpro.2024.109613
9. M. A. Kouritzin, Markov observation models and deepfakes, *Mathematics*, **13** (2025), 2128. https://doi.org/10.3390/math13132128
10. M. A. Kouritzin, Residual and stratified branching particle filters, *Comput. Stat. Data Anal.*, **111** (2017), 145–165. https://doi.org/10.1016/j.csda.2017.02.003
11. W. Lenz, Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern, *Z. Phys.*, **21** (1920), 613–615.

12. V. Maroulas, A. Nebenführ, Tracking rapid intracellular movements: a Bayesian random set approach, *Ann. Appl. Stat.*, **9** (2015), 926–949. https://doi.org/10.1214/15-AOAS819

13. M. Oren, M. Hassid, N. Yarden, Y. Adi, R. Schwartz, Transformers are multi-state RNNs, *arXiv Preprint*, 2024. https://doi.org/10.48550/arXiv.2401.06104

14. A. M. Schäfer, H. G. Zimmermann, Recurrent neural networks are universal approximators, *Int. J. Neural Syst.*, **17** (2007), 253–263. https://doi.org/10.1142/S0129065707001111

AIMS Press