

動的ボルツマンマシンの連続拡張

梶野 洸*1

Hirosi Kajino

*1日本アイ・ピー・エム株式会社東京基礎研究所

IBM Research - Tokyo

We present a functional dynamic Boltzmann machine (F-DyBM), which is an extension of a dynamic Boltzmann machine (DyBM) for a functional time series. We also present an online learning algorithm for F-DyBM. The effectiveness of F-DyBM is confirmed by a numerical experiment.

1. 序論

動的ボルツマンマシン (dynamic Boltzmann machine, DyBM) [Osogami 15] はニューラルネットを用いた多次元時系列の生成モデルである。適格度トレースを用いて過去の時系列の要約統計量を逐次的に計算することで比較的長期間の時間依存性を考慮できる点と時系列を遡る誤差逆伝播法をすることなくオンラインに学習できるという点で、リカレントニューラルネットワーク (RNN) [Rumelhart 86] などの他の手法と差別化される。

本研究では、動的ボルツマンマシンを部分観測関数時系列に適用するための連続拡張を提案する。ここで、関数時系列とは各離散時刻 $t \in \mathbb{Z}$ に関数 $f^{[t]}: \mathcal{X} \rightarrow \mathbb{R}$ が観測される時系列を指し、各時刻に有限観測点 (点数や観測位置は各時刻変化し得る) での関数の評価値が観測される関数時系列を部分観測関数時系列と定義する。部分観測関数時系列の例として時空間データが挙げられる。携帯型デバイスのセンサデータを用いて将来を予測する問題を考えると、 $f^{[t]}(x)$ は場所 x 時刻 t でのセンサの値 (気温, 水質, 汚染物質濃度など) に相当する。このとき $f^{[t]}(x)$ は任意の地点 $x \in \mathcal{X}$ に定義されるため関数時系列となるが、携帯型デバイスは有限個しかなくその位置や個数が変わりうることから、得られるデータは部分観測関数時系列となる。このようなデータが各時刻得られた元で将来の関数を予測する問題を取り扱う。

2. 記法と問題設定

本論文では次に述べるような記法を用いる。関数 $f: \mathbb{R}^D \rightarrow \mathbb{R}$ と行列 $X = [x_1 \ \dots \ x_N]^\top \in \mathbb{R}^{N \times D}$ に対して、 $f(X) := [f(x_1) \ \dots \ f(x_N)]^\top \in \mathbb{R}^N$ と定義する。関数 $K: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ と二つの行列 $X, Y = [y_1 \ \dots \ y_M]^\top \in \mathbb{R}^{M \times D}$ に対して、 $K(X, Y)$ を (n, m) 成分が $K(x_n, y_m)$ となるサイズ $N \times M$ の行列とする。時刻 t に関する値を $f^{[t]}$ としたときに、時刻 $-\infty$ から $t-1$ までの値の系列を $f^{[<t]}$ と書く。

次に本論文で取り扱う問題を定義する。観測を行う空間を \mathcal{X} とする。時空間データの場合 $\mathcal{X} \subset \mathbb{R}^2$ または $\mathcal{X} \subset \mathbb{R}^3$ である。各時刻 $t \in \mathbb{Z}$ に定義される関数 $f^{[t]}$ をパターンと定義し、 $f^{[t]}(x) \in \mathbb{R}$ を $x \in \mathcal{X}$ で観測される信号を定義する。本論文で扱う問題は、各時刻 t にパターン $f^{[t]}$ の有限個の信号が観測され

たときに、次の時刻のパターン $f^{[t+1]}$ を予測するという問題である。時刻 t での観測点の集合を $X^{[t]} = [x_1^{[t]} \ \dots \ x_{N^{[t]}}^{[t]}]^\top$ と置く。観測点の位置とその個数 $N^{[t]}$ は時刻に依存して変化する。

3. 動的ボルツマンマシン

はじめに既存研究である動的ボルツマンマシン, 特に多次元実数時系列を取り扱うガウス型動的ボルツマンマシン (Gaussian DyBM, G-DyBM) [Dasgupta 17, Osogami 16] を紹介する。

G-DyBM は N 個のニューロンを用いて N 次元時系列をモデル化する。各時刻観測される N 次元のパターン $\mathbf{f}^{[t]} = [f_1^{[t]} \ \dots \ f_N^{[t]}]^\top$ の各次元 n の信号 $f_n^{[t]} \in \mathbb{R}$ をニューロン n が出力するとモデル化する。G-DyBM はこの時系列を、

$$p(\mathbf{f}^{[t]} | \mathbf{f}^{[<t]}) = \prod_{n=1}^N \mathcal{N}(f_n^{[t]}; \mu_n^{[t]}, \sigma_n^2), \quad (1)$$

$$\mu^{[t]} = \mathbf{b} + \sum_{\delta=1}^d W^{[\delta]} \mathbf{f}^{[t-\delta]} + \sum_{l=1}^L U_l \alpha_l^{[t-1]}, \quad (2)$$

$$\alpha_l^{[t-1]} = \sum_{\delta=1}^{\infty} \lambda_l^{\delta-1} \mathbf{f}^{[t-\delta]} \quad (l = 1, \dots, L), \quad (3)$$

とモデル化する。ここでモデルパラメタは \mathbf{b} , $\{W^{[\delta]}\}_{\delta=1}^d$, $\{U_l\}_{l=1}^L$ で、ハイパーパラメタは $d, L, \{\sigma_n\}_{n=1}^N, \{\lambda_l\}_{l=1}^L$ である。時刻 t のパターンは平均 $\mu^{[t]} = [\mu_1^{[t]} \ \dots \ \mu_N^{[t]}]^\top$ を経由して過去すべてのパターンに依存して生成される (式 (1))。時刻 t の平均は二種類のメモリを用いて計算される (式 (2))。ひとつは直近の d 個のパターン $\{\mathbf{f}^{[t-\delta]}\}_{\delta=1}^d$ を記憶するメモリで、もうひとつは適格トレースと呼ばれるメモリ $\alpha_l^{[t-1]}$ ($l = 1, \dots, L$) で過去すべてのパターンを要約したものとなっている (式 (3))。

4. 関数動的ボルツマンマシン

関数時系列を取り扱うことができる動的ボルツマンマシンとして関数動的ボルツマンマシン (functional DyBM, F-DyBM) を提案する。従来の動的ボルツマンマシンでは有限個のニューロンを用いて多次元時系列をモデル化していたのに対して、関数動的ボルツマンマシンでは \mathcal{X} に連続的に分布したニューロンを用いて関数時系列をモデル化する。また部分観測関数時系列から F-DyBM を学習するオンラインアルゴリズムを提案する。

連絡先: 梶野 洸, KAJINO@jp.ibm.com

4.1 モデル

G-DyBM では各ニューロンはインデックス n と結びついていて、F-DyBM では各ニューロンは各地点 $x \in \mathcal{X}$ と結びついている。時刻 t でのパタンを $f^{[t]}: \mathcal{X} \rightarrow \mathbb{R}$ とし、地点 x のニューロンの信号を $f^{[t]}(x)$ とする。G-DyBM のモデルを定義する各式 (1), (2), (3) をそれぞれ式 (4), (5), (6) のように拡張することで F-DyBM を定義する。

式 (1) は、ガウス分布の代わりにガウス過程を用いることで

$$p(f^{[t]} | f^{[<t]}) = \mathcal{GP}(f^{[t]}; \mu^{[t]}, K_{\sigma^2}), \quad (4)$$

のように拡張される。ここで $f^{[t]}$ は時刻 t のパタン、 $\mu^{[t]}: \mathcal{X} \rightarrow \mathbb{R}$ は時刻 t の平均関数、 $K_{\sigma^2}: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ は共分散関数^{*1} である。G-DyBM と同様に、 $f^{[t]}$ は $\mu^{[t]}$ を通じて過去すべてのパタン $f^{[<t]}$ に依存する。また式 (2), (3) は、

$$\begin{aligned} \mu^{[t]}(x) = & b(x) + \sum_{\delta=1}^d \int_{\mathcal{X}} w^{[\delta]}(x, x') f^{[t-\delta]}(x') dx' \\ & + \sum_{l=1}^L \int_{\mathcal{X}} u_l(x, x') \alpha_l^{[t-1]}(x') dx', \end{aligned} \quad (5)$$

$$\alpha_l^{[t-1]}(x) = \sum_{\delta=1}^{\infty} \lambda_l^{\delta-1} f^{[t-\delta]}(x), \quad (6)$$

と拡張される。ここで、重み行列は重み関数で置き換え、行列演算は積分で置き換えた。モデルパラメタは $b(x)$, $\{w^{[\delta]}(x, x')\}_{\delta=1}^d$, $\{u_l(x, x')\}_{l=1}^L$ で、ハイパーパラメタは d , L , $\{\lambda_l\}_{l=1}^L$, K_{σ^2} である。

4.2 オンライン学習アルゴリズム

前節では G-DyBM を拡張する形で F-DyBM を導出した。本節では、F-DyBM のモデルパラメタを部分観測関数時系列からオンライン学習する手法を提案する。このような手法を構築する上で、以下の三つの課題を解決する必要がある。

課題 (i) モデルパラメタ b , $w^{[\delta]}$, u_l や関数時系列はそれぞれ関数であるため、有限のメモリに保存できる形で表す必要がある。

課題 (ii) 式 (5) を計算するために重み関数と過去のパタンの積分計算が必要があるが、数値積分は計算コストが高いため工夫が必要である。

課題 (iii) 入力に関数時系列が部分観測であるため、それを直接 F-DyBM の学習に用いることができない。

4.2.1 導出

上記の課題を解決するために、まず重み関数を有限個のパラメタでモデル化する。特に、ガウス過程の共分散関数である K_{σ^2} を用いて重み関数 $w^{[\delta]}(x, x')$, $u_l(x, x')$, $b(x)$ を

$$w^{[\delta]}(x, x') = K_{\sigma^2}(x, P) W^{[\delta]} K_{\sigma^2}(P, x') \quad (\delta = 1, \dots, d), \quad (7)$$

$$u_l(x, x') = K_{\sigma^2}(x, P) U_l K_{\sigma^2}(P, x') \quad (l = 1, \dots, L), \quad (8)$$

$$b(x) = K_{\sigma^2}(x, P) \mathbf{b}, \quad (9)$$

とモデル化する。ここで $P = [p_1 \ \dots \ p_N]^\top$ を \mathcal{X} 内の点の集合とする。このモデル化により、F-DyBM のモデルパラ

*1 クロネッカーのデルタ $\delta_{x, x'}$ を用いて $K_{\sigma^2}(x, x') := K(x, x') + \sigma^2 \delta_{x, x'}$ と定義する。

Algorithm 1 オンライン学習アルゴリズム

```

1: for  $t = 0, 1, 2, \dots$  do
2:   時刻  $t$  のパタンを  $\mu^{[t]}(x)$  (式 (10)) と予測する。
3:   時刻  $t$  のパタンを地点  $X^{[t]}$  で観測し  $f^{[t]}(X^{[t]})$  を得る。
4:   パラメタを式 (13)–(16) のように更新する。
5:    $\hat{f}^{[t]}(P)$  を式 (12) のように計算する。
6:    $Q$  と  $\mathcal{E}$  を  $\hat{f}^{[t]}(P)$  を使って更新する。
7: end for

```

メタは $W^{[\delta]}, U_l \in \mathbb{R}^{N \times M}$, $\mathbf{b} \in \mathbb{R}^N$ となる。さらに式 (7), (8), (9) を式 (5) に代入すると、

$$\begin{aligned} \mu^{[t]}(x) = & K_{\sigma^2}(x, P) \left[\mathbf{b} + \sum_{\delta=1}^d W^{[\delta]} f^{[t-\delta]}(P) \right. \\ & \left. + \sum_{l=1}^L U_l \alpha_l^{[t-1]}(P) \right], \end{aligned} \quad (10)$$

が得られる。重み関数のモデル化に共分散関数を使ったため、積分を解析的に計算することができる。このようにして、課題 (i), (ii) は解決された。

上記のモデル化による他の効果として、関数時系列の一部のみ記憶すれば学習・予測を行う上では十分となることがあげられる。式 (10) の計算のためには、固定された点集合 P での観測値のみ記憶すればよく、入力として与えられる時変な観測点集合 $X^{[t]}$ 上の観測値を記憶する必要がない。この特徴は、適格トレースを定義する上で特に有用である。実際、適格トレース $\alpha_l^{[t-1]}(P)$ は式 (3) と同じように、

$$\alpha_l^{[t-1]}(P) = \sum_{\delta=1}^{\infty} \lambda_l^{\delta-1} f^{[t-\delta]}(P), \quad (11)$$

と定義することができ、各時刻 $f^{[t]}(P)$ が得られるとすると G-DyBM と同様に再帰的に更新可能であるため、一定量のメモリで過去の時系列を要約したものとなっている。

次に課題 (iii) を解決するために、式 (10), (11) に現れる関数パタンを推定量で逐次的に置き換える。各時刻 t に $f^{[t]}(X^{[t]})$ と $\mu^{[t]}(x)$ が得られている元で $f^{[t]}(P)$ の最大事後確率推定量は、

$$\hat{f}^{[t]}(P) = \mu^{[t]}(P) + K(P, X^{[t]}) K_{\sigma^2}(X^{[t]}, X^{[t]})^{-1} d^{[t]}(X^{[t]}), \quad (12)$$

である。ここで $d^{[t]}(x) := f^{[t]}(x) - \mu^{[t]}(x)$ と置いた。各時刻 t で、最大事後確率推定量 $\hat{f}^{[t]}(P)$ を式 (12) のように計算し、それを時刻 $t+1$ で降式 (10) の $f^{[t]}(P)$ の代わりとして使うことで課題 (iii) を解決する。

4.2.2 アルゴリズム

前節で説明した手法を用いて F-DyBM のオンライン学習アルゴリズムをアルゴリズム 1 に示す。これはモデルパラメタ $\Theta = \{W^{[\delta]}\}_{\delta=1}^d \cup \{U_l\}_{l=1}^L \cup \{\mathbf{b}\}$ をオンラインに学習し、一時刻先のパタンの予測を与える。アルゴリズムの入力は、F-DyBM のハイパーパラメタ $d, L, P \in \mathbb{R}^{N \times D}$, K_{σ^2} , $\{\lambda_l\}_{l=1}^L$ である。はじめに二つのメモリ Q, \mathcal{E} とモデルパラメタ Θ を初期化する。ここで Q は $\{\hat{f}^{[t-\delta]}(P)\}_{\delta=1}^d$ を記憶するキューで、 \mathcal{E} は適格トレースの集合 $\{\alpha_l^{[t-1]}(P)\}_{l=1}^L$ である。各時刻 $t = 0, 1, 2, \dots$ で、まず時刻 t のパタンを式 (10) のように予

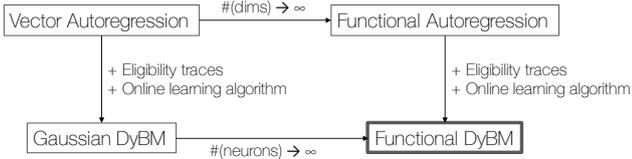


図 1: VAR, FAR, G-DyBM, F-DyBM の関係図.

測する．次に $(X^{[t]}, f^{[t]}(X^{[t]}))$ を観測し, モデルパラメタ Θ を,

$$\Theta \leftarrow \Theta + \eta^{[t]} \cdot \frac{\partial \mathcal{L}^{[t]}(\Theta)}{\partial \Theta}, \quad (13)$$

と更新する．ここで $\eta^{[t]}$ は学習率, $\mathcal{L}^{[t]}(\Theta)$ はさきほど得られた観測の条件付き対数尤度で, Θ に依存しない定数 C を用いて,

$$\begin{aligned} \mathcal{L}^{[t]}(\Theta) &:= \log p(f^{[t]}(X^{[t]}) | f^{[<t]}; \Theta) \\ &= -\frac{1}{2} d^{[t]}(X^{[t]})^\top K_{\sigma^2}(X^{[t]}, X^{[t]})^{-1} d^{[t]}(X^{[t]}) + C, \end{aligned}$$

と書ける．各モデルパラメタに関する $\mathcal{L}^{[t]}(\Theta)$ の勾配は,

$$\frac{\partial}{\partial \mathbf{b}} \mathcal{L}^{[t]}(\Theta) = K(P, X^{[t]}) K_{\sigma^2}(X^{[t]}, X^{[t]})^{-1} d^{[t]}(X^{[t]}), \quad (14)$$

$$\frac{\partial}{\partial W^{[\delta]}} \mathcal{L}^{[t]}(\Theta) = \left[\frac{\partial}{\partial \mathbf{b}} \mathcal{L}^{[t]}(\Theta) \right] \hat{f}^{[t-\delta]}(P)^\top, \quad (15)$$

$$\frac{\partial}{\partial U_l} \mathcal{L}^{[t]}(\Theta) = \left[\frac{\partial}{\partial \mathbf{b}} \mathcal{L}^{[t]}(\Theta) \right] \alpha_l^{[t-1]}(P)^\top, \quad (16)$$

と書ける．パラメタ更新後, 二つのメモリ \mathcal{Q}, \mathcal{E} を更新する．そのためにまず $\hat{f}^{[t]}(P)$ を式 (12) のように計算する． $\hat{f}^{[t]}(P)$ を \mathcal{Q} に挿入し, 最後の要素を取り除くことで \mathcal{Q} を更新する． \mathcal{E} は,

$$\alpha_l^{[t]}(P) = \lambda_l \alpha_l^{[t-1]}(P) + \hat{f}^{[t]}(P) \quad (l = 1, \dots, L), \quad (17)$$

のように更新する．

5. 数値実験

関連手法と比べると, F-DyBM のモデルは適格トレースで過去の時系列の情報を活用している点 (特徴 (i)) と関数時系列を直接モデル化している点 (特徴 (ii)) が特徴である．本実験ではその二つの効果を定量的に評価することを目的とする．

5.1 比較手法

F-DyBM の二つの特徴をそれぞれを抜いたモデルと比較を行う．F-DyBM と比較に用いた手法の関係を図 1 に示す．F-DyBM から特徴 (i) を抜いたモデルは関数自己回帰モデル (Functional Autoregression, FAR) [Bosq 00] と呼ばれる．F-DyBM から特徴 (ii) を抜いたモデルは G-DyBM である．また F-DyBM から特徴 (i), (ii) を抜いたモデルはベクトル自己回帰 (Vector Autoregression, VAR) [Lütkepohl 05] と呼ばれる．VAR と FAR は通常オフラインで学習をするが, 本論文で取り扱う問題設定に適用するため G-DyBM や F-DyBM のようなオンライン学習アルゴリズムを適用する．また VAR

と G-DyBM は直接関数時系列を取り扱うことはできないため, 次に述べる補間法を組合せて本問題設定に適用する．

VAR と G-DyBM で関数時系列を取り扱うために, まず各ニューロンを \mathcal{X} の地点と関連付ける． n 個目のニューロンが地点 $p_n \in \mathcal{X}$ にあるとし, 地点の集合を $P = [p_1 \dots p_N]^\top$ と書く．各時刻の関数パターン $f^{[t]}$ がガウス過程 $\mathcal{GP}(0, K_{\sigma^2})$ に従っているとすると, その部分観測 $f^{[t]}(X^{[t]})$ が得られた元で, P での信号は,

$$\hat{f}^{[t]}(P) = K(P, X^{[t]}) K_{\sigma^2}(X^{[t]}, X^{[t]})^{-1} f^{[t]}(X^{[t]}),$$

と推定できる．このように推定した信号を VAR や G-DyBM の観測信号とみなすことで VAR や G-DyBM を学習させる．また予測についても同様になると, VAR や G-DyBM の予測 $\mu^{[t]}$ が得られた元で, 任意の地点 $x \in \mathcal{X}$ での予測は,

$$\hat{f}^{[t]}(x) = K(x, P) K_{\sigma^2}(P, P)^{-1} \mu^{[t]},$$

と推定できる．

5.2 データセット

性能比較のために五つの実データセットを用いる．特に序論で時空間データに関する事例を紹介したため, 気温と汚染物質濃度に関する時空間データを用いる．

5.2.1 NOAA Global Surface Temperature V4.01

NOAA が提供している気温データを用いる．1880 年 1 月から現在までの世界中の気温異常度^{*2}を記録したデータで, 長さ 1,638 の時系列である．地球を $5^\circ \times 5^\circ$ 単位で区切った点での観測結果が記録されている．時系列の初めから終わりまで一貫して観測値が存在する 392 地点を実験に用いた．各観測地点について $x \in \mathbb{R}^3$ は, 地球を単位球とみなしたときの対応する点のユークリッド座標を用いた．

5.2.2 AirData

気温データに加え, AirData^{*3} から取得した大気中の四つの汚染物質の濃度に関するデータを用いる．基準ガスである CO, NO₂, O₃, SO₂ の一日ごとの濃度を 1990 年から 2016 年 11 月まで抽出して使用した．濃度データはアメリカ合衆国の屋外観測地点で記録された．

このデータセットに対し次のような前処理を施した．各データセットについて, National Ambient Air Quality Standards で規定されている primary standard のデータを抽出し, Event Type = None に該当するデータのみを使用した．これは野火などによる異常データを除外するためである．次に各観測日, 各観測地点について重複しているデータが存在するならば一番はじめのデータを抽出した．最後に, 各観測値 $c \in \mathbb{R}_+$ を $\log(c + 10^{-5})$ と変換し, その値を $f^{[t]}(x)$ と定義した．ここで, t は日にちに相当し, $x \in \mathbb{R}^3$ は観測地点に相当する． x の定義は NOAA データセットと同様である．各データセットは長さ 9,831 で, 時間によって観測地点の数 $N^{[t]}$ が変わる．CO, NO₂, O₃, SO₂ の $N^{[t]}$ の平均と標準偏差はそれぞれ 401.3 ± 87.6 , 371.0 ± 37.8 , 870.0 ± 260.9 , 540.7 ± 107.5 である．

*2 <http://www.esrl.noaa.gov/psd/data/gridded/data.noaaglobaltemp.html> から 2016 年 8 月 23 日にダウンロードした．

気温異常度は, 長期平均などの標準値からの解離度を表している．

*3 AirData は the United States Environmental Protection Agency から提供されており, 今回使用したデータセットは Air Quality System Data Mart (http://aqsdrr1.epa.gov/aqswweb/aqstmp/airdata/download_files.html) から 2016 年 12 月 28 日に取得した．

表 1: テストセットでの RMSE の平均および標準偏差 .

	NOAA	CO	NO ₂	O ₃	SO ₂
VAR	1.164 ± 0.001	2.184 ± 0.068	1.723 ± 0.040	1.383 ± 0.042	3.931 ± 0.018
FAR	1.158 ± 0.003	1.880 ± 0.124	1.668 ± 0.067	0.561 ± 0.024	3.888 ± 0.042
G-DyBM	1.155 ± 0.003	1.872 ± 0.099	1.604 ± 0.047	0.598 ± 0.056	3.852 ± 0.034
F-DyBM	1.131 ± 0.009	1.768 ± 0.073	1.520 ± 0.014	0.457 ± 0.049	3.750 ± 0.058

5.3 実験手順

次に性能評価のために用いた実験手順を説明する．データセット, 手法に関わらず同じ手順を用いる．はじめに複数のハイパーパラメタ設定を用意する．この中から良い設定を選びその汎化性能を測りたい．時系列を 3 : 3 : 4 の比率で時間順に訓練セット, 検証セット, テストセットに分割する．各ハイパーパラメタ設定で訓練セットを用いてオンライン学習アルゴリズムを適用する．それぞれの訓練されたモデルの性能を検証セット上で測定する．性能は, 各時刻で得られる一時刻先の予測と真のデータの検証データ全体での二乗平均平方根誤差 (RMSE) を計算することで測る．各ハイパーパラメタ設定で測った性能のうち, 最も RMSE が小さいものを選び, 検証セットと同様にしてテストセットを用いて性能を測り, それをその手法の性能とした．今回の設定では, 各手法は訓練セット, 検証セット, テストセットそれぞれ一周のみ用い, また検証セットやテストセットでもオンライン学習を行っている．

カーネルは RBF カーネル $K(x, x'; \gamma) = \exp(-\gamma \|x - x'\|^2)$ を用い, ハイパーパラメタ設定は以下の範囲のすべての組合せを用いた． $N = 25, d = 3, \eta^{[0]} \in \{2^{-n}\}_{n=19}^{23}, \sigma^2 \in \{2^{-n}\}_{n=0}^2, \gamma \in \{2^{-n}\}_{n=0}^5$ をすべてのモデルに用い, G-DyBM と F-DyBM については $L = 3, \lambda_1 = 0.1, \lambda_2 = 0.5, \lambda_3 = 0.9$ を追加で用いた．また G-DyBM と VAR では $\sigma_n = 1.0 (n = 1, \dots, N)$ とした．VAR の検証スコアが他手法と比べて大きく劣ることがあったため, VAR については $\gamma \in \{2^{-n}\}_{n=-3}^{-1}$ を追加した．いずれの手法も SGD でパラメタを更新し, 学習率は rmsprop [Tieleman 12] を用いて調整した． $P \in \mathbb{R}^{N \times 3}$ の各点は $[-1, 1]^3$ 上の一様分布から独立に生成した後, 単位ベクトルに正規化した． P の生成に乱数を用いるため, 上記の実験を 10 回繰り返しその平均・標準偏差を報告する．

5.4 結果

実験結果を表 1 に示す．はじめに特徴 (i) である適格トレースの効果を調べる．F-DyBM の RMSE は FAR の RMSE の 81–98% (平均 92%) であり, G-DyBM の RMSE は VAR の RMSE の 43–99% (平均 84%) であることから, 適格トレースはいずれのデータセットにおいても性能向上に寄与することがわかる．また特徴 (ii) である関数時系列の直接的なモデル化についても同様に性能向上に寄与するといえる．具体的には, F-DyBM の RMSE は G-DyBM の RMSE の 76–98% (平均 92%) であり, FAR の RMSE は VAR の RMSE の 41–99% (平均 84%) である．以上より, 本実験の範囲ではいずれの特徴も性能向上に寄与することが示された．特徴 (i) を加えることで平均 12.0% RMSE を下げることができ, 特徴 (ii) を加えることで平均 11.7% RMSE を下げることができる．

6. 結論

本論文では, ガウ斯的動的ボルツマシン (G-DyBM) の拡張として関数動的ボルツマシン (F-DyBM) を導出し, そのオンライン学習アルゴリズムを提案した．モデルの導出は比較的容

易にすることができる一方で, オンライン学習アルゴリズムを構成するためには三つの課題を解決する必要があった．それは, モデルに関数パラメタや関数値観測が含まれておりそのままでは数値計算で扱えないこと, モデルに積分が含まれること, 関数時系列が部分観測であることである．二つの解決法を組合せてこれらの課題を解決した．まず関数パラメタを有限個のパラメタとカーネル関数を組合せてモデル化することをはじめ二つの課題を解決した．また有限地点の観測から関数時系列を逐次的に推定し, 更新式にプラグインすることで最後の課題に対処した．この二つの方法を組合せることで F-DyBM のオンライン学習アルゴリズムを構成することができた．関連手法と比べて F-DyBM は適格トレースを用いて過去の時系列を記憶していることと関数時系列を直接モデル化をしている点で特徴的であるため, その二つの特徴の効果を数値実験で調査した．その結果, 適格トレースにより誤差が平均 12.0% 減少し, 関数時系列のモデル化により誤差が 11.7% 減少した．以上より, この二つの特徴を兼ね備えた F-DyBM の有用性が検証された．

参考文献

- [Bosq 00] Bosq, D.: *Linear Processes in Function Spaces: Theory and Applications*, chapter 5, Springer New York (2000)
- [Dasgupta 17] Dasgupta, S. and Osogami, T.: Nonlinear dynamic Boltzmann machines for time-series prediction, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (2017)
- [Lütkepohl 05] Lütkepohl, H.: *New Introduction to Multiple Time Series Analysis*, Part I, Springer Berlin Heidelberg (2005)
- [Osogami 15] Osogami, T. and Otsuka, M.: Seven neurons memorizing sequences of alphabetical images via spike-timing dependent plasticity, *Scientific Reports*, Vol. 5, pp. 14149 EP – (2015)
- [Osogami 16] Osogami, T.: Learning binary or real-valued time-series via spike-timing dependent plasticity, in *The first NIPS Workshop Computing with Spikes* (2016)
- [Rumelhart 86] Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: *Learning internal representations by error propagation*, chapter 8, pp. 318–362, MIT Press (1986)
- [Tieleman 12] Tieleman, T. and Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSE: Neural Networks for Machine Learning (2012)