周期構造を取り込んだ深層確率時系列モデル

Deep Stochastic Time Series Model Incorporated with Periodic Structure

数学専攻　朝倉　剛一
ASAKURA, Koichi

1　はじめに

近年, デジタル化の進展に伴い, 様々な時系列データが収集, 活用されるようになった．そのため, 時系列データに対する統計的モデリングの重要性が高まっている．時系列データに対して, これまで用いられてきた統計モデルを総称して古典的な時系列モデルと呼ぶ．古典的な時系列モデルは周期構造や, 外部要因といったデータの特徴に合わせてモデル化できるため, モデルの解釈性に優れている．しかし, 古典的な時系列モデルは高次元データやビッグデータに対して, 各成分を関連付けながら同時にモデル化できない, 分析者がデータから読み取れることしかモデル化できないという 2 つの欠点がある．対して, 多層のニューラルネットを用いた機械学習の方法論である深層学習を用いると, この課題を解決できる．しかし, 深層学習はブラックボックス化して出力するため, 分析者は結果を解釈することが難しい．

そこで近年, 古典的な時系列モデルをベースに, 深層学習を用いるモデルが多く提案されている．このモデルを総称して, 深層確率時系列モデル (Deep Stochastic Time Series Model) と呼ぶ．深層確率時系列モデルは上記の 2 つよりもモデルが柔軟で, 予測の精度が高い．また, 深層学習に比べてモデルの解釈性が上がっている．

2　状態空間モデル

古典的な時系列モデルの例として, データの背後にある特徴を潜在変数として表現する状態空間モデル (State Space Model) がある．時刻 t における N 次元の観測値時系列を $\boldsymbol{x}_t = \left(x_t^{(1)}, \dots, x_t^{(N)}\right)^{\mathrm{T}} \in \mathbb{R}^N$ とする．ここで, $x_t^{(i)} \in \mathbb{R},\ i = 1, \dots, N$ である．このとき, 線形ガウス状態空間モデルは $t = 1, \dots, T$ に対して,

$$\boldsymbol{x}_t = \boldsymbol{A}_t \boldsymbol{z}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim N(\boldsymbol{0}, \boldsymbol{E}_t), \tag{2.1}$$

$$\boldsymbol{z}_t = \boldsymbol{F}_t \boldsymbol{z}_{t-1} + \boldsymbol{R}_t \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\boldsymbol{0}, \boldsymbol{Q}_t) \tag{2.2}$$

と定義され, 観測方程式 (2.1) と状態方程式 (2.2) からなる．L 次元の $\boldsymbol{z}_t \in \mathbb{R}^L$ は潜在的な確率変数で, 状態と呼ばれる．初期状態は $\boldsymbol{z}_0 \sim N(\boldsymbol{y}_0, \boldsymbol{P}_0)$ と仮定する．観測値 $\boldsymbol{x}_t$ は状態 $\boldsymbol{z}_t$ と, 正規ホワイトノイズである観測値撹乱項 $\boldsymbol{\varepsilon}_t \in \mathbb{R}^N$ によって変動する．また, 状態 $\boldsymbol{z}_t$ は 1 時点前の状態 $\boldsymbol{z}_{t-1}$ と, 正規ホワイトノイズである状態撹乱項 $\boldsymbol{\eta}_t \in \mathbb{R}^R$ によって変動する．

観測値 $x_t^{(i)}$ に関する状態と観測方程式 (2.1), 状態方程式 (2.2) の係数行列を

$$\boldsymbol{A}_t^{(i)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}, \quad \boldsymbol{z}_t^{(i)} = \begin{pmatrix} z_t^{\mathrm{season}^{(i)}} \\ z_{t-1}^{\mathrm{season}^{(i)}} \\ \vdots \\ z_{t-p+2}^{\mathrm{season}^{(i)}} \end{pmatrix}, \quad \boldsymbol{F}_t^{(i)} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ 1 & & & \\ & \ddots & & O \\ O & & 1 & \end{pmatrix}, \quad \boldsymbol{R}_t^{(i)} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

と定義すると, 状態 $\boldsymbol{z}_t^{(i)}$ は季節性を表現するモデルになる．このモデルを周期 p の季節調整モデルという．

3　深層学習

状態空間モデルは周期構造や, 外部要因といったデータの特徴に合わせてモデル化できるため, モデルの解釈性に優れている．しかし, 高次元データに対して不向きで, トレンドや季節性以外の分析者には気付けない潜在要因を考慮できずにモデル化してしまう．そこで, 深層学習を用いてモデル化することで, この課題を解決できる．ここで概説する深層学習は, 時系列データに対して用いる再帰型ニューラルネットワーク (Recurrent Neural Network, RNN) と, データから特徴を抽出できる変分自己符号化器 (Variational Auto-Encoder, VAE) である．

3.1　再帰型ニューラルネットワーク

RNN によって, 時系列の入力データ $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ から出力 $\boldsymbol{y}_t$ を予測する．ニューラルネットワークは入力層, 隠れ層, 出力層から構成されており, RNN は隠れ層の内部に閉路を持つニューラルネットワークの総称である．この構造のおかげで情報を一時的に記憶し, 振る舞いを動的に変化させることができる．RNN の動作は各時刻 t につき 1 つの入力 $\boldsymbol{x}_t$ を受け取り, 同時に 1 つの出力 $\boldsymbol{y}_t$ を返すというものである．

時刻 t における入力層の出力を $\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,I})^{\mathrm{T}}$, 隠れ層の入力を $\boldsymbol{a}_t = (a_{t,1}, \ldots, a_{t,H})^{\mathrm{T}}$, 出力を $\boldsymbol{h}_t = (h_{t,1}, \ldots, h_{t,H})^{\mathrm{T}}$, 出力層の入力を $\boldsymbol{v}_t = (v_{t,1}, \ldots, v_{t,K})^{\mathrm{T}}$, 出力を $\boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,K})^{\mathrm{T}}$ とする．また, 入力層と隠れ層間の重みを $\boldsymbol{W}^{(\mathrm{in})}$, 隠れ層から隠れ層への帰還路の結合の重みを $\boldsymbol{W}^{(\mathrm{hidden})}$, 隠れ層と出力層間の重みを $\boldsymbol{W}^{(\mathrm{out})}$ と表記する．

このとき, 時刻 t における隠れ層の入力 $\boldsymbol{a}_t$ は同時刻 t における入力層の出力 $\boldsymbol{x}_t$ と, 帰還路によって伝播される時刻 $t-1$ における隠れ層の出力 $\boldsymbol{h}_{t-1}$ から得られる．この入力 $\boldsymbol{a}_t$ から活性化関数 $\boldsymbol{f}^{(\mathrm{hidden})}$ を経由して, 隠れ層の出力 $\boldsymbol{h}_t$ が得られる．つまり, 隠れ層の入力と出力はそれぞれ

$$\boldsymbol{a}_t = \boldsymbol{W}^{(\mathrm{in})}\boldsymbol{x}_t + \boldsymbol{W}^{(\mathrm{hidden})}\boldsymbol{h}_{t-1}, \qquad \boldsymbol{h}_t = \boldsymbol{f}^{(\mathrm{hidden})}(\boldsymbol{a}_t)$$

となる．

続いて, 時刻 t における出力層の入力 $\boldsymbol{v}_t$ は同時刻 t における隠れ層の出力 $\boldsymbol{h}_t$ から得られる．この入力 $\boldsymbol{v}_t$ から活性化関数 $\boldsymbol{f}^{(\mathrm{out})}$ を経由して, 出力層の出力 $\boldsymbol{y}_t$ が得られる．つまり, 出力層の入力と出力はそれぞれ

$$\boldsymbol{v}_t = \boldsymbol{W}^{(\mathrm{out})}\boldsymbol{h}_t, \qquad \boldsymbol{y}_t = \boldsymbol{f}^{(\mathrm{out})}(\boldsymbol{v}_t)$$

となる．以上のように, ネットワーク内部にある帰還路によって出力を計算することで, RNN は過去に受け取った全ての入力を出力に影響させる．

3.2　変分自己符号化器

VAE は高次元入力 $\boldsymbol{x}$ のデータ分布の表現をエンコーダで行い, 低次元の潜在変数 $\boldsymbol{z}$ に埋め込む．デコーダで $\boldsymbol{z}$ から $\boldsymbol{x}$ を導出する関数に近似的に等しい分布からデータ $\tilde{\boldsymbol{x}}$ を生成する．潜在変数 $\boldsymbol{z}$ の次元はハイパーパラメータである．

VAE では一般的にデータ $\boldsymbol{x}$ が正規分布に従うと仮定する． したがって, デコーダは $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = N\left(\boldsymbol{x}\,\middle|\,\boldsymbol{\mu}^{\mathrm{dec}}, \boldsymbol{\Sigma}^{\mathrm{dec}}\right)$ である．この分布のパラメータ $\boldsymbol{\mu}^{\mathrm{dec}}$ と $\boldsymbol{\Sigma}^{\mathrm{dec}}$ はパラメータが θ, 入力が $\boldsymbol{z}$, 出力が $\boldsymbol{\mu}^{\mathrm{dec}}$ と $\boldsymbol{\Sigma}^{\mathrm{dec}}$ の深層ニューラルネットワークから, $\left(\boldsymbol{\mu}^{\mathrm{dec}}, \boldsymbol{\Sigma}^{\mathrm{dec}}\right) = \mathrm{NN}_\theta^{\mathrm{dec}}(\boldsymbol{z})$ と得られる．ここで, $\mathrm{NN}_\theta^{\mathrm{dec}}$ はパラメー

タが θ のニューラルネットワークで, デコーダを表す．ゆえに, 生成モデルは同時分布 $p_\theta(\boldsymbol{x},\boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})$ となる．分布 $p_\theta(\boldsymbol{z})$ は事前分布であり, 多変量正規分布 $N\left(\boldsymbol{z}\left|\boldsymbol{\mu}^{\mathrm{prior}},\boldsymbol{\Sigma}^{\mathrm{prior}}\right.\right)$ である．ハイパーパラメータは一般的に $\left(\boldsymbol{\mu}^{\mathrm{prior}},\boldsymbol{\Sigma}^{\mathrm{prior}}\right) = (\boldsymbol{0},\boldsymbol{I})$ とする．

続いて, 事後分布 $p(\boldsymbol{z}|\boldsymbol{x})$ を考える．一般的に, 事後分布 $p(\boldsymbol{z}|\boldsymbol{x})$ は複雑になるため, 学習における計算が解析的にできない．そこで事後分布をパラメトリックな分布 $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = N\left(\boldsymbol{z}\left|\boldsymbol{\mu}^{\mathrm{enc}},\boldsymbol{\Sigma}^{\mathrm{enc}}\right.\right)$ によって近似する．この分布のパラメータは, 深層ニューラルネットワーク $(\boldsymbol{\mu}^{\mathrm{enc}},\boldsymbol{\Sigma}^{\mathrm{enc}}) = \mathrm{NN}_\phi^{\mathrm{enc}}(\boldsymbol{x})$ から得られる．ここで, $\mathrm{NN}_\phi^{\mathrm{enc}}$ はパラメータが ϕ のニューラルネットワークで, エンコーダを表す．

4　深層確率時系列モデル

深層学習はモデルや出力結果を解釈できないという欠点がある．そこで, 古典的な時系列モデルをベースに, 深層学習を用いる深層確率時系列モデルが多く提案されている．その中でも, 深層学習を用いて状態空間モデルの観測方程式と状態方程式を表現できる Neural Kalman Dynamic Model (DynaNet) [1] を説明する．

4.1　DynaNet: Neural Kalman Dynamic Model

DynaNet は線形ガウス状態空間モデルの状態方程式と観測方程式を VAE で表現する．まず, エンコーダ $\mathrm{NN}_\phi^{\mathrm{enc}}$ を用いて, 時刻 t でデータ $\boldsymbol{x}_t$ から特徴 $\boldsymbol{z}_t$ と観測値撹乱項の係数行列 $\boldsymbol{G}_t$ を抽出する．つまり,

$$(\boldsymbol{z}_t,\boldsymbol{G}_t) = \mathrm{NN}_\phi^{\mathrm{enc}}(\boldsymbol{x}_t)$$

である．続いて, 1 時点前にカルマンフィルタ [3] で導出されたフィルタ化推定量 $\boldsymbol{y}_{t-1|t-1} = \mathrm{E}(\boldsymbol{z}_{t-1}|\boldsymbol{x}_{1:t-1})$ を入力として, LSTM [2] から遷移行列 $\boldsymbol{F}_t$ と状態撹乱項の係数行列 $\boldsymbol{R}_t$ を生成する．つまり,

$$(\boldsymbol{F}_t,\boldsymbol{R}_t) = \mathrm{LSTM}\left(\boldsymbol{y}_{t-1|t-1}\right)$$

である．次に, エンコーダから生成した潜在変数と状態空間モデルの各係数行列を用いて, カルマンフィルタから時刻 t のフィルタ化推定量 $\boldsymbol{y}_{t|t} = \mathrm{E}(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$ を更新する．

最後に, フィルタ化推定量 $\boldsymbol{y}_{t|t}$ を入力値とするデコーダから, 同時刻の目的値を

$$\tilde{\boldsymbol{x}}_t = \mathrm{NN}_\theta^{\mathrm{dec}}\left(\boldsymbol{y}_{t|t}\right)$$

と出力する． 同時刻の観測値が欠測していて $\boldsymbol{z}_t$ が利用できない場合, 過去の 1 期先予測推定量 $\boldsymbol{y}_{t|t-1} = \mathrm{E}(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})$ から

$$\hat{\boldsymbol{x}}_t = \mathrm{NN}_\theta^{\mathrm{dec}}\left(\boldsymbol{y}_{t|t-1}\right)$$

と推定される．

モデルのパラメータ θ,ϕ は, 平均二乗誤差関数を導出し, 誤差関数を最小化することで学習される．この誤差関数は真の値 $\boldsymbol{x}_t$ と, 予測値 $\tilde{\boldsymbol{x}}_t,\hat{\boldsymbol{x}}_t$ の差を同時にとって,

$$\mathcal{L}(\theta,\phi) = \frac{1}{T}\sum_{t=1}^{T}\left(\|\boldsymbol{x}_t-\tilde{\boldsymbol{x}}_t\|^2 + \|\boldsymbol{x}_t-\hat{\boldsymbol{x}}_t\|^2\right)$$

となる．誤差関数からパラメータに関する勾配を計算し, 誤差逆伝播法を用いて勾配を更新する．続いて, 更新した勾配から Adam [4] を用いてパラメータの学習をする．

4.2 周期構造を取り込んだ DynaNet

深層確率時系列モデルは VAE によって, 時系列データの特徴を表す潜在変数を構成した．しかし, この潜在変数は何を表すのか解釈することができない．すなわち, 依然として深層確率時系列モデルは解釈性に乏しい．そこで, 深層確率時系列モデルに周期構造を取り入れることで, 解釈性の向上が期待できる．

時刻 t における観測値の潜在変数である状態を $\boldsymbol{z}_t = \left(\boldsymbol{z}_t^{\text{season}}, \boldsymbol{z}_t^{\text{other}}\right)^{\mathrm{T}}$ と表す．ここで, $\boldsymbol{z}_t^{\text{season}}$ は季節性を表す潜在変数, $\boldsymbol{z}_t^{\text{other}}$ は VAE により導出される解釈できない潜在変数である．このとき, 周期構造を取り込んだ DynaNet を提案する．状態方程式の各係数行列は

$$
\boldsymbol{F}_t = \begin{pmatrix} \boldsymbol{F}_t^{\text{season}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{F}_t^{\text{other}} \end{pmatrix}, \quad \boldsymbol{F}_t^{\text{season}} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ 1 & & & \\ & \ddots & & O \\ O & & 1 & \end{pmatrix},
$$

$$
\boldsymbol{R}_t = \begin{pmatrix} \boldsymbol{R}_t^{\text{season}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{R}_t^{\text{other}} \end{pmatrix}, \quad \boldsymbol{R}_t^{\text{season}} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}
$$

であり, $\boldsymbol{F}_t^{\text{other}}, \boldsymbol{R}_t^{\text{other}}$ は 1 時点前の潜在状態 $\boldsymbol{z}_{t-1}$ のフィルタ化推定量 $\boldsymbol{y}_{t-1|t-1}$ を入力とする LSTM から生成される．また, VAE のエンコーダは観測値 $\boldsymbol{x}_t$ を入力として, 潜在状態 $\boldsymbol{z}_t$ と観測方程式のノイズ項の係数行列 $\boldsymbol{G}_t$ を生成する．観測値 $\boldsymbol{x}_t$ を生成するデコーダは, VAE のデコーダによってカルマンフィルタから生成されたフィルタ化推定量 $\boldsymbol{y}_{t|t}$ を入力として導出される．パラメータ θ, ϕ の学習方法は DynaNet と同様である．

5 おわりに

本研究では, 時系列データに対する深層学習を用いたモデルとして, 近年研究が進んでいる深層確率時系列モデルを整理した．また, いくつかの深層確率時系列モデルに対して, 周期構造の取り込みを中心に拡張した．

今後の課題として 2 点あげる．1 つ目は, 多くの提案がされている深層確率時系列モデルの中で, どのモデルが最適なモデルなのか判断できないことである．様々なデータに対して, 分析による各モデルの比較が必要である．2 つ目は依然として, 深層確率時系列モデルの解釈が困難なことである．VAE から生成した潜在変数 $\boldsymbol{z}$ が何を表すのか解釈する方法を模索する必要がある．

参考文献

[1] Chen, C., Lu, C. X., Wang, B., Trigoni, N. and Markham, A. (2019). DynaNet: Neural Kalman dynamical model for motion estimation and prediction, *arXiv preprint, arXiv:*1908.03918.

[2] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Computation*, 9(8), 1735-1780.

[3] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, 82(1), 35-45.

[4] Kingma, D. P. and Ba, J. L. (2014). Adam: A method for stochastic optimization, *arXiv preprint, arXiv:*1412.6980.