マルチタスク・ベイズ的最適化を用いた複数の時系列データの分析と予測

中山　峻一†　　江口　浩二††

† 神戸大学大学院システム情報学研究科　〒 657–8501 兵庫県神戸市灘区六甲台町 1-1
†† 神戸大学大学院システム情報学研究科　〒 657–8501 兵庫県神戸市灘区六甲台町 1-1
E-mail: †snpc94@cs25.scitec.kobe-u.ac.jp, ††eguchi@port.kobe-u.ac.jp

あらまし　近年時系列データ分析の重要性が高まっており，その中でも，ディープラーニングをはじめとした機械学習を用いて時系列データ分析を行う研究に注目が集まっている．機械学習を用いる際，学習率などの超パラメータを適切に設定することができるかどうかが機械学習の性能を決定する重要な要素になっている．本稿では，複数の時系列データの関連性に着目したマルチタスク・ベイズ的最適化を行う．中でも，パラメータのサンプリングに着目したマルチタスク・ベイズ的最適化を長短期メモリネットワークの超パラメータ探索に適用し，複数の株価指数の予測問題に対して提案手法の有効性の評価を行う．

キーワード　深層学習，時系列データ分析，長短期メモリネットワーク，ベイズ的最適化，超パラメータ探索

1. はじめに

近年時系列データ分析の重要性が高まっており，様々な手法によってデータに含まれている潜在的な特徴を抽出して活用する試みが多くなされている．その中でも，ディープラーニングをはじめとした機械学習を用いて時系列データ分析を行う研究に注目が集まっている．機械学習の適用範囲は多岐にわたり，画像データや音声データなど枚挙にいとまがないが，時系列データ分析においても一定の成果を挙げている．代表的なものに自己回帰モデル（ARIMA）や長短期メモリネットワーク（Long Short-Term Memory：LSTM）がある．

機械学習を用いる際，学習率などの超パラメータを適切に設定することができるかどうかが機械学習の性能を決定する重要な要素になっている．従来，超パラメータを決定する技術としてしばしば格子探索法やランダム探索法が用いられてきたが，機械学習が扱う問題の規模や機械学習モデルの複雑さが拡大するにつれて，これらの技術をもってしても探索の性能および時間において十分な結果が得られなくなる可能性がある．

最近の研究の動向として，大域的な最適化に一般的に用いられているベイズ的最適化（Bayesian Optimization）を超パラメータ探索に適用する考えが検討されている．この方法は探索と活用のトレードオフを上手く設定することができ，多様なモデルに対してよい性能を示している．さらに文献 [1] や [2] などで複数の類似したデータセットを想定して互いの探索結果を活用するマルチタスク・ベイズ的最適化（Multi-Task Bayesian Optimization）についての研究も活発に行われており，いずれも単一データセットのベイズ的最適化を上回る結果を得ている．　本稿では複数の株価指数の時系列データに対して分析及び予測を行う回帰問題に対して，再帰型ニューラルネットワークモデルの 1 つである LSTM [3] を適用した際の超パラメータ探索について検討する．文献 [1] では Branin-Hoo 関数 [4] や MNIST データセットを用いたロジスティック回帰，CIFAR-10 [5] データセットを用いた畳み込みニューラルネットワーク（Convolutional neural networks: CNN）に対して，文献 [2] では Support Vector Machine（SVM）に対してマルチタスク・ベイズ的最適化を用いて超パラメータ最適化を行う実験を行っているが，再帰型ニューラルネットワークに対して行われたベイズ的最適化に関する研究事例は我々の知る限りない．また，株価指数はその国の経済状況を表すものである為，経済状況が類似する株価指数は相互に関連するデータセットであると考えることができる．従って，そのような時系列データに対する再帰型ニューラルネットワークの超パラメータ探索手法としてマルチタスク・ベイズ的最適化が有効であると期待される．本稿の目的は複数の時系列データの関連性に着目したマルチタスク・ベイズ的最適化を用いて，LSTM に対して最適なパラメータの値の組を行うことにある．

マルチタスク・ベイズ的最適化の手法としては，マルチタスク・ガウス過程（Multi-Task Gaussian Process）を用いる手法 [1] やメタ特徴（Meta feature）を用いるもの [2] があるが，本稿では後者，とりわけ超パラメータのサンプリングに着目したマルチタスク・ベイズ的最適化を行う．すなわち学習対象とする株価指数と関連すると推測される株価指数の時系列データセットを用い、その最適化結果の一部を関連するデータセットのサンプリングとするマルチタスク・ベイズ的最適化を実現する．

2. 関連研究

ここでは関連研究として，従来から時系列データ分析の際によく用いられる ARIMA モデル，本稿において事前学習として用いる Deep Belief Network(DBN)，および近年，時系列データ分析に用いられることの多い再帰型ニューラルネットワーク，とりわけ LSTM について説明する．

2.1 自己回帰和分移動平均モデル

自己回帰和分移動平均モデル（ARIMA モデル）[6] とは，ある系列 y_t が非定常モデルであり、かつ差分系列 $\Delta y_t = y_t - y_{t-1}$ が定常モデルとなるような過程（これを I(1) 過程と呼ぶ）に対

して，その差分系列が定常かつ反転可能な自己回帰移動平均モデル（ARMA モデル）のことを指す．

ここで，過程が定常であるとは過程が持つ期待値と自己共分散が時間を通して一定であるということである．これより，定常な過程はトレンドを持たない過程であるといえる．また，自己回帰移動平均モデルは移動平均モデル（MA）モデルと自己回帰モデル（AR）モデルを組み合わせたモデルを指す．以下では MA モデルおよび AR モデル，および ARMA モデルの概要を示す．

MA モデルはホワイトノイズを拡張したものであり，以下のようなホワイトノイズの線形和で表すことができる．(MA(q) 過程)

$$y_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q}, \quad \varepsilon \sim W.N.(\sigma^2) \tag{1}$$

ここで μ は期待値であり，θ はパラメータである．また，$W.N.$ はホワイトノイズのことを指し，ε は σ^2 のホワイトノイズに従う．MA(q) モデルによって，q 次元の自己相関モデルを表現することができる．しかし，MA(q) モデルはモデル化するために q このパラメータが必要であり，パラメータはデータから推測しなければならない．また，このモデルは観測できないホワイトノイズの線形和で表されるためにモデルの解釈が難しく，推定や予測が困難であるという問題がある．この問題に対して次に紹介する AR モデルによって回避できる可能性がある．AR モデルは，過程が自身の過去に回帰された形で表現される過程である．p 次 AR 過程（AR(p) 過程）は以下のようになる．

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t, \quad \varepsilon \sim W.N.(\sigma^2) \tag{2}$$

ここで c は定数であり，ϕ はパラメータである．経済データは循環的な変動を示す傾向があり，AR モデルを用いることで循環的な自己相関を記述することができる．

これらの 2 つのモデルの項を組み合わせたモデルが自己回帰移動平均モデル（ARMA モデル）である．形式的には以下のように表される．

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q}, \quad \varepsilon \sim W.N.(\sigma^2) \tag{3}$$

AR モデルと MA モデルの持つ性質のうち，有する性質が強い方が ARMA モデルの性質となる．

2.2 DBN(Deep Belief Network)

本節では DBN の仕組みにおいて基礎となる Restricted Boltzmann Machine(RBM) についての説明を行い，その後 DBN の概要について説明を行う．

2.2.1 RBM(Restricted Boltzmann Machine)

RBM(Restricted Boltzmann Machine) とは，生成モデルの中で最も基礎的なものの 1 つである．生成モデルとはネットワークの振る舞いを確率分布によって表現するようなモデルであり，出力が確率分布によって決定される点が通常の順伝播型ネットワークと異なる点である．層の構成は，可視層と呼ばれる入力データに直接寄与する層と潜在層と呼ばれる入力データから潜在的に表現される特徴を持つ 2 種類の 2 値ユニットの層からなる．また，同一層間の各ユニットにおける結合は存在していないという制約を持つ．n_v 個の 2 値確率変数を持つ可視層を $\boldsymbol{v}$，n_h 個の 2 値確率変数を持つ可視層を $\boldsymbol{h}$ と定義する．この時，RBM はエネルギー関数 $E(\boldsymbol{v}, \boldsymbol{h})$ を用いて次式のように表現することができる．

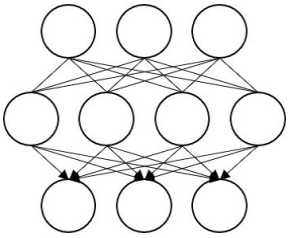

図 1　Deep Belief Network(DBN).

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z}\exp\{-E(\boldsymbol{v}, \boldsymbol{h})\} \tag{4}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{h} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{h} \tag{5}$$

$$Z = \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}}\exp\{E(\boldsymbol{v}, \boldsymbol{h})\} \tag{6}$$

ここで，Z は分配関数と呼ばれる正規化定数である．分配関数においては $\boldsymbol{v}, \boldsymbol{h}$ の状態を網羅的に計算する必要があり，演算が困難であるという問題を抱えている．また，正規化された結合確率分布 $P(\boldsymbol{v})$ を扱うことが困難である事も知られている．

ただし，RBM の条件付き分布 $P(\boldsymbol{h}|\boldsymbol{v})$，$P(\boldsymbol{v}|\boldsymbol{h})$ は因子性を持つという特徴を有し，計算および標本化が可能である．このため，Contrastive Divergence 法（CD 法）と呼ばれるギブスサンプリングから端を発したサンプリング手法を用いることで上記の問題を解決している．条件付き分布 $P(\boldsymbol{h}|\boldsymbol{v})$，$P(\boldsymbol{v}|\boldsymbol{h})$ の計算法は文献 [7]，Contrastive Divergence 法の詳細については文献 [8] を参照してほしい．

2.2.2 DBN(Deep Belief Network)

DBN（Deep Belief Network）[7] とは，Hinton らが 2006 年に開発した初期の生成モデルの 1 つであり，現在の深層学習の流行の火付け役となったモデルである．

DBN は RBM を複数段重ねた形で構成され，最上位の 2 層間の接続は無向グラフによる接続であり，その他の層は有効グラフによる結合となっている．図 1 に中間層が 1 層の場合の DBN の構成例を示す．元となる制約ボルツマンマシンは可視層と潜在層は共に 2 値の確率変数によって構成されるものであるが，DBN においては入力層として扱う可視層は 2 値または実数値を取ることが可能である．上に述べた，中間層が 1 層の DBN の確率分布は以下のようになる．

$$P(\boldsymbol{h}^{(l)}, \boldsymbol{h}^{(l-1)}) \propto \exp\left(\boldsymbol{b}^{(l)^{\mathrm{T}}}\boldsymbol{v}^{(l)} + \boldsymbol{b}^{(l-1)^{\mathrm{T}}}\boldsymbol{h}^{(l-1)} + \boldsymbol{v}^{(l-1)^{\mathrm{T}}}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)}\right) \tag{7}$$

$$P(h_i^{(k)} = 1|\boldsymbol{h}^{(k+1)}) = \sigma\left(b_i^{(k)} + \boldsymbol{W}_{:,i}^{(k+1)^{\mathrm{T}}}\boldsymbol{h}^{(k+1)}\right) \quad \forall i, \forall_k \in 1, \ldots, l-2 \tag{8}$$

$$P(v_i = 1|\boldsymbol{h}^{(1)}) = \sigma\left(b_i^{(0)} + \boldsymbol{W}_{:,i}^{(1)^{\mathrm{T}}}\boldsymbol{h}^{(1)}\right) \tag{9}$$

可視層が実数値を取る場合は，3 番目の式の代わりに扱いやすくするために対角行列 $\boldsymbol{\beta}$ を用いた以下の式を使用する．

$$v \sim \mathcal{N}\left(\boldsymbol{v}; \boldsymbol{b}^{(0)} + \boldsymbol{W}^{(1)^{\mathrm{T}}}\boldsymbol{h}^{(1)}, \boldsymbol{\beta}^{-1}\right) \tag{10}$$

DBN から標本化を行う際には，最上位の 2 層の間でギブスサンプリングを行い，単一経路で伝播させることで可視層からサンプルを抽出することが可能となる．

DBN の学習は CD 法または最尤推定法が用いて以下の手順で行われる．まず，1 層目の RBM において対数尤度 $\mathbb{E}_{\boldsymbol{v}\sim p_{data}} \log p(\boldsymbol{v})$ を最大化するように学習を行う．次に，2 層目の RBM を対数尤度

$$\mathbb{E}_{\boldsymbol{v}\sim p_{data}} \mathbb{E}_{\boldsymbol{h}^{(1)}\sim p^{(1)}(\boldsymbol{h}^{(1)}|\boldsymbol{v})} \log p^{(2)}(\boldsymbol{h}^{(1)}) \tag{11}$$

を最大化するように学習する．ここで，$p^{(1)}, p^{(2)}$ はそれぞれ 1 層目，2 層目の RBM によって表現される確率分布である．2 層目の RBM は 1 層目の潜在ユニットを標本化することによって定義される分布をモデル化するように学習される．この手順を繰り返すことで DBN の学習が行われる．

DBN は主に特徴抽出器として使用されることが多く、文献 [7] においては MLP（多層パーセプトロン）に対して DBN の生成的学習を介して得られた重み及びバイアスでサンプリングするファインチューニングが行われている．また，最上位の層の更に上に出力層を定義し，誤差逆伝播法などを用いてモデルの調整を行うことで教師あり学習を行うことも可能である．

2.3 再帰型ニューラルネットワーク

2.3.1 再帰型ニューラルネットワーク

再帰型ニューラルネットワーク（Recurrent Neural Network：RNN）[9] とは，ネットワーク内部に有向閉路を持つことにより，再帰的な処理を可能にしたニューラルネットワークの総称である．閉路を持つことにより学習した情報を一時的に記憶し，処理ごとに異なる振る舞いを可能にしている．これにより系列を持つデータ間の関連性を推察し，従来の順伝播型ニューラルネットワークと比較して分類問題の効率的な処理が可能となった．図 2 に再帰型ニューラルネットワークの概要を示す．

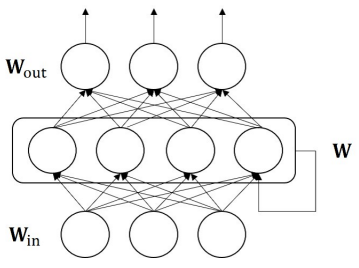

図 2　Recurrent Neural Network(RNN).

ここで，$\mathbf{W}_{in}$ は入力層，$\mathbf{W}_{out}$ は出力層，および $\mathbf{W}$ は中間層を表している．中間層が自分自身への帰還路を有していること以外は従来の順伝播型ニューラルネットワークと同等である．

2.3.2 LSTM（Long Short-Term Memory）

再帰型ニューラルネットワークでは理論上では過去の入力情報を一時的に記憶することが可能である．しかし実際には層の数が多いニューラルネットワークでは勾配降下法を用いて逆誤差伝播法を行う際，勾配が消失あるいは発散してしまう性質がある．このため実際に記憶できるのは高々10 ステップ程度の記憶にとどまると言われている．

LSTM [3] はこの問題を解決するために，ニューラルネットワークの中間層をメモリユニットと呼ばれる構造を設定し，入力情報の中・長期記憶を可能とした．図 3 にメモリユニットの概略を示す．

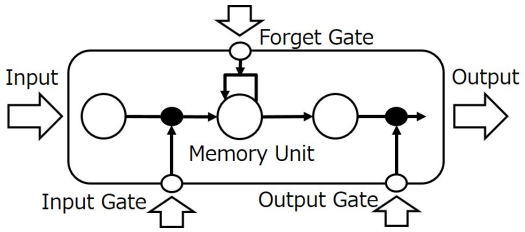

図 3　LSTM memory units.

ここで,「Input Gate」 では 「Input」 から得られる情報を「MeMory Cell」 に入力するかどうかを決める役割,「Output Gate」 では 「Memory Cell」 から得られる情報を 「Output」 に出力するかどうかを決める役割を果たしている．

3. マルチタスク・ベイズ的最適化による時系列モデルの超パラメータ探索

3.1 ベイズ的最適化

本節ではマルチタスク・ベイズ的最適化と基となるベイズ的最適化の概要について述べる．

ベイズ的最適化とは，計算的に高コストでかつ非凸であるかブラックボックスである関数 f に対して最適化を行う際に用いられる大域的最適化手法の一つである．パラメータ最適化問題は大域的最適化が求められる状況の 1 つである．ベイズ的最適化の主な戦略としては，関数そのものを直接的に決定して最適化するのではなく，関数が比較的コストの低い確率モデル（確率過程）に従うという利用を利用して最適化を行う点にある．

n 番目のデータを $\mathcal{D}_n = \{\mathbf{x}_n, y_n\}_{n=1}^N$ とし，入力を $\mathbf{x}_n = \{x_1, ..., x_m\}$（ここで m は超パラメータの数），出力を y_n は出力を意味するもとのする．超パラメータ最適化においては，入力は各超パラメータの設定値の集合，出力は入力を関数 f に与えた際の評価値を指す．評価値の形式としては，損失（loss）や正解率（accuracy），検証誤差（validation error）などがある．本稿では検証誤差を採用する．

以上を踏まえると，形式的にベイズ的最適化は超パラメータを対象とした場合では以下のように述べることができる．まず，ベイズ的最適化は逐次探索アルゴリズムであるため，n 番目のデータおよびその際の超パラメータ探索を用いる必要があ

る．そして，出力がこれまでの探索結果の中で最良のものよりも良くなるように $n+1$ 番目の超パラメータ探索を決める必要がある．その際にベイズ的最適化では，獲得関数（acquisition function）$\alpha(\mathbf{x};\mathcal{D}_n)$ を用いて最大化する必要がある．すなわち，以下の式のようにして $n+1$ 番目の超パラメータ探索 $\mathbf{x}_{n+1}$ を決定すればよい．

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}}{\operatorname{argmax}}\, \alpha(\mathbf{x};\mathcal{D}_n) \tag{12}$$

具体的な獲得関数の例については文献 [10] を見てほしい．

次に，獲得関数によって選ばれた超パラメータ値の組 $\mathbf{x}_{n+1}$ を関数 f に適用し，誤差 y_{n+1} を得る．ここで，超パラメータ最適化問題の場合にはデータセット $\mathcal{D}_n$ から関数を定式化することは計算コストの観点からも容易ではないことが多い．そこで，$\mathcal{D}_n$ が何かしらの確率モデルに従っていると仮定をおくことによって回帰を用いて y_{n+1} を得ることができる．一般的には確率モデルにはガウス過程 [11] が用いられることが多い．

その後，ベイズの定理を用いて，事前分布 $p(\mathcal{D}_n|(\mathbf{x}_{n+1},y_{n+1}))$ から事後分布 $p((\mathbf{x}_{n+1},y_{n+1})|\mathcal{D}_{n+1})$ を推定する．これによって $n+1$ 番目のデータ $\mathcal{D}_{n+1}$ が生成されると仮定する．形式的には以下の式のように記述できる．

$$\begin{aligned} &p((\mathbf{x}_{n+1},y_{n+1})|\mathcal{D}_{n+1}) \\ &\quad = \frac{p(\mathbf{x}_{n+1},y_{n+1})p(\mathcal{D}_n|(\mathbf{x}_{n+1},y_{n+1}))}{p(\mathcal{D}_n)} \end{aligned} \tag{13}$$

そして最後に，超パラメータ探索 $\mathbf{x}_{n+1}$ と誤差 y_{n+1}，およびデータ $\mathcal{D}_{n+1}$ を用いて確率モデル $\mathcal{M}$ を更新する．この節で述べたベイズ的最適化の手続きはアルゴリズム 1 に擬似コードが示されている．

Algorithm 1 Bayesian Optimization

for $n = 1, 2, ...,$ **do**
  select new $\mathbf{x}_{n+1}$ by maximize acquisition function α
  $\mathbf{x}_{n+1} = \operatorname{argmax}_{\mathbf{x}} \alpha(\mathbf{x};\mathcal{D}_n)$
  obtain y_{n+1}
  add data $\mathcal{D}_{n+1}$ by use Bayes' theorem
  $p((\mathbf{x}_{n+1},y_{n+1})|\mathcal{D}_{n+1}) = \frac{p(\mathbf{x}_{n+1},y_{n+1})p(\mathcal{D}_n|(\mathbf{x}_{n+1},y_{n+1}))}{p(\mathcal{D}_n)}$
  update probabilistic model $\mathcal{M}$
end for

ベイズ的最適化は D_n の情報を基に最適化を行うため、D_n の初期設定を決定する必要がある．そこで，我々はベイズ的最適化の試行回数のうち，数回はランダムに $\mathbf{x}$ を決定する処理を行う．本稿ではこの処理を「サンプリング」と定義する．4 章で述べる実験においては 3 個の超パラメータの値の組を取得している．

3.2 マルチタスク・ベイズ的最適化

ベイズ的最適化は探索と活用のトレードオフをうまく設定できる手法であったために，多くの機械学習を用いた最適化問題において格子探索法およびランダム探索法に対してよりよい性能を発揮していることが示されている．しかし，ベイズ的最適化には「コールドスタート問題」と呼ばれる問題が存在する．これは，新しいデータセットを用いたデータセットが与えられるたびに一から最適化を実行しなければならないことを指している．これはデータセットに類似性がある場合においても同様である．手動で決定する手法ならば感覚的に類似性を見抜いて効率よく探索することができる可能性があるにもかかわらず，ベイズ的最適化ではその機会を逃しているといえる．

この問題を解決する方法として，データセット間におけるデータセットの類似性を考慮したマルチタスク・ベイズ的最適化（Multi-task Bayesian Optimization）が近年, 複数の研究者によって提案されている．代表的な手法としては文献 [2] で述べられているようなメタ特徴（Meta Feature）を用いて関連するデータセットの最適化結果を初期値にするマルチタスク・ベイズ的最適化と，文献 [1] に述べられているようなデータセット間の類似性をカーネル関数としてガウス過程に組み込んだマルチタスク・ガウス過程を用いる手法がある．本稿では文献 [2] の手法について概要を述べる．

文献 [2] で述べられている手法は，これから機械学習を用いて超パラメータ探索を行う新たなデータセット D_{N+1} に対して，D_{N+1} に関連するデータセット群 $D_i(i=1,...,N)$ の中から D_{N+1} に最も類似性が高いデータセットの $\mathbf{x}^*$ を D_{N+1} に対する初期値とする考え方である．

データセット間の類似度を評価する指数として，文献 [2] では 2 つの指数が紹介されている．1 つ目は一般的に広く用いられている L_p ノルムである．L_p ノルムを以下に示す．

$$d(D_i,D_j) = \|\mathbf{m}^i - \mathbf{m}^j\|_p \tag{14}$$

通常は $p=1$ あるいは $p=2$ とする．この評価指数を用いるには，予めデータセット $D_i(i=1,...,N)$ ごとにメタ特徴 $\mathbf{m}^i=(m_1^i,...,m_n^i)$ を計算する必要がある．文献 [2] では 5 つのグループに大別される 46 のメタ特徴を実装し，計算している．

2 つ目は異なる超パラメータ探索ごとの性能の類似性に着目した指数である．すなわち，D_i, D_j の両方のデータセットにそれぞれ存在する n 個の超パラメータを用い，D_i, D_j を別々に順位付けした結果 $F^{D_i}=[f^{D_i}(\mathbf{x}_1),...,f^{D_i}(\mathbf{x}_n)]$ および $F^{D_j}=[f^{D_j}(\mathbf{x}_1),...,f^{D_j}(\mathbf{x}_n)]$ 用いて求めるスピアマン相関係数のことである．形式的には次のような式になる．

$$\begin{aligned} d(D_i,D_j) &= 1 - Corr(F^{D_i},F^{D_j}) \\ Corr(F^{D_i},F^{D_j}) &= \frac{6\sum_{k=1}^{n} d_k^2}{n^3-n} \end{aligned} \tag{15}$$

ここで，d_k は F^{D_i} における k 番目のデータの順位 $f^{D_i}(\mathbf{x}_k)$ と F^{D_j} における k 番目のデータの順位 $f^{D_j}(\mathbf{x}_k)$ の差である．

ただし，この指数はデータセット D_{n+1} に対する関数の評価値は得られていないので $f^{D_{N+1}}(\mathbf{x}_1),...,f^{D_{N+1}}(\mathbf{x}_n)$ を評価することはできない．近似的に求める方法として，D_{n+1} においても計算可能なメタ特徴の組 $\langle \mathbf{m}^i, \mathbf{m}^i \rangle$ を $d(D_i,D_j)$ に写像する関数 R を回帰を用いて学習することにより，以下の式のように相関係数を決定する方法がある．

$$d(D_{N+1},D_j) \approx R(\mathbf{m}^{N+1},\mathbf{m}^j) \tag{16}$$

$\mathbf{x}_n(n=1,...,N)$ をデータセット $D_n(n=1,...,N)$ に対す

る最良の超パラメータ探索とすると，すでに述べた評価指数を用いて初期探索アルゴリズムは次のように記述できる．まず，評価指数 d を D_{N+1} と $D_i(i = 1, ..., N)$ に対してそれぞれ計算しこの値を $\phi(i) = d(D_{N+1}, D_i)$ とする．次に，$\phi(i)$ を評価指数の値が増加する順にソートする．そして，t 個だけ解 $\mathbf{x}^{\phi(1)}, ..., \mathbf{x}^{\phi(t)}$ を取り出す．特に，$t = 1$ の場合は評価指数と最も近いデータに対する初期値が得られることになる．擬似コードはアルゴリズム 2 に示されている．

Algorithm 2 Initialization

```
sort dataset by increasing distance to D_{N+1} i.e.:
(φ(i) ≦ φ(j)) ⇔ (d(D_{N+1}, D_i) ≦ d(D_{N+1}, D_j))
for i = 1, ..., t do
  x_i = x^{φ(i)}
end for
x* = BayesianOptimization(⟨x_1, ..., x_t⟩)
return x*
```

本稿で用いるマルチタスク・ベイズ的最適化は，上記で述べた手法における前半部分の「関連するデータセットの最適化結果を初期値にする」部分に焦点を当てたものである．

3.3 マルチタスク・ベイズ的最適化の LSTM への適用

本稿では前節で述べた文献 [2] で行われている最適化手法を基にして，関連するデータセットの最適化結果から得られる超パラメータの値の組の一部を次のデータセットのサンプリングされた値とするマルチタスク・ベイズ的最適化を行っている．具体的には以下の手順でマルチタスク・ベイズ的最適化を行っている．

(1). 目的のデータセットと関連のあるデータセットに対して，通常のベイズ的最適化を行う

(2). サンプリングを行う回数だけ、誤差の小さい順に超パラメータ値の組を抽出する

(3). 目的のデータセットでベイズ的最適化を行う際に，2. で得られた超パラメータ値の組をサンプリングされた値として使用する

(4). 目的のデータセットでベイズ的最適化を行う

関連するデータセットが複数ある場合は，目的のデータセットに至るまで上記のアルゴリズムを繰り返し適用することにより最適化を行うことが可能となる．以上の手順の概略図を図 4 に示す．

この例においては正解率（accuracy）を評価値としておき，値が高い順に関連データセット（relational task）におけるパラメータの値の組をランク付けしている．そしてその中で正解率が高い上位 3 つのパラメータ値の組を目的データセット（objective task）のサンプルとしている．このようにサンプリングに着目することでデータセットの関連度が高い場合には従来のベイズ的最適化に比べてより早い段階から最適値付近の領域を探索することが可能となる．

また，本稿においては前章で述べたように LSTM を問題を解くためのモデルとして採用する．LSTM は以下のような超パラメータを有する．

（1） 重みの更新に関する超パラメータ：学習率，ϵ，ρ，バッチサイズ，エポック数

（2） モデルの超パラメータ：LSTM の層の数，ドロップアウト率，ユニット数

本稿では学習アルゴリズムとして RMSprop を用いている．RMSprop は再帰型ニューラルネットワークにおいて定評のある学習アルゴリズムであり，形式的には以下のような式になる．

$$\begin{aligned} h_t &= \alpha h_{t-1} + (1-\rho)\nabla E(\boldsymbol{w}^t)^2 \\ \eta_t &= \frac{\eta_0}{\sqrt{h_t} + \epsilon} \\ \boldsymbol{w}^{t+1} &= \boldsymbol{w}^t - \eta_t \nabla E(\boldsymbol{w}^t) \end{aligned} \tag{17}$$

ここで，$\boldsymbol{w}$ は重みであり，η は学習率である．RMSprop の詳細は文献 [12] を参照してほしい．

また，バッチサイズ及びエポック数はミニバッチ学習に関する超パラメータであり，LSTM の層の数は LSTM を何層積み重ねるかを決める超パラメータである．つまり，本研究の実験で用いる LSTM は通常の LSTM を拡張した深層 LSTM（Deep-LSTM）であるといえる．さらに，ドロップアウト率は Dropout と呼ばれる過学習を防ぐための超パラメータであり，ユニット数は各 LSTM 層ごとの出力次元の大きさである．これらの超パラメータの全てを対象とし，本節で述べたマルチタスク・ベイズ的最適化を行った．実験の詳細に関しては次章で述べる．

4. 実　　験

4.1 使用するデータセット

本稿の実験においては，2008 年 7 月 1 日から 2016 年 9 月 30 日までの 1 日単位の期間の国ごとの株価指数データを使用する．使用したデータセットは WIND database から得られるものであり，5 カ国 6 種類の株価指数に関するデータが含まれている．

本稿では，以下の 5 カ国の株価指数のデータを使用している．なお，括弧内は国と開始年を示す．

- S&P 500 （United States，1923）
- Nikkei 225 （Japan，1950）
- Hang Seng （Hong-Kong，1969）
- CSI 300 （China，2005）
- Nifty 50 （India，2009）

前章の 3.3 節において述べたマルチタスク・ベイズ的最適化手法において，鍵となる要素は関連するデータセットを実行する順番にある．例えば GPS の軌跡から移動手段を予測する問題の場合は，調査対象領域に隣接する領域においてベイズ的最適化を行った結果を用いて，上記のマルチタスク・ベイズ的最適化を行うとが望ましいと考えられる．本稿で用いるデータセットに関しては開始年の古い指数ほど先進国に近いと考え，株価指数が制定された開始年の古い順でデータセットを行うことに

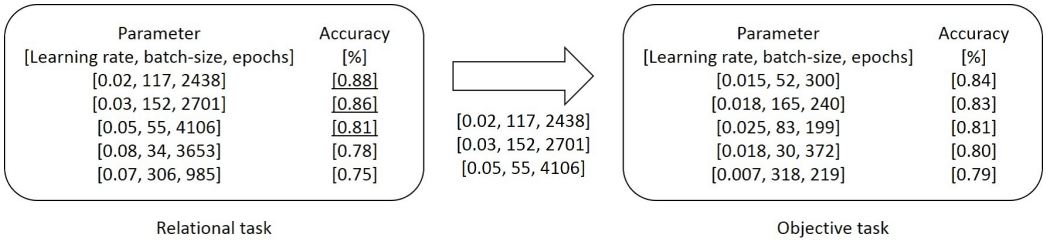

図 4　The outline of Multi-Task Bayesian Optimization for LSTM.

した．

また，各指数には表 1 のような目的変数及び説明変数が存在する．

表 1　Description of the input variables.

取引データ	テクニカル指数	その他
始値，終値，高値 安値，取引量	MACD，CCI，ATR， BOLL，EMA20，MA5/MA10 MTM6/MTM12，SMI， ROC，WVAD	取引レート

本稿では目的変数として終値（Closed Price）を用い，説明変数として表 1 にある変数のすべてを用いる．

4.2　データセットの前処理

本稿においては実験をより適切に行うために，前節で述べたデータセットに対して以下のような前処理を行った．

まず，2 年 6ヶ月分のデータセットを 1 つの組として 2 年分を訓練データ，3ヶ月分を検証データ，残りのデータをテストデータとする．検証データを用いることにより，汎化性能（未知のデータに対する予測性能）を客観的に判断することができる．

1 つの組に対するデータセットの学習が終えると，次は以前のデータセットよりも 3ヶ月先の時点から 2 年 6ヶ月分のデータセットを一つの組として取得する．この時，モデルは前のデータセットですでに学習されたものを用いる．すなわち，モデルは事前学習が行われた状態になっている．上記の操作をテストデータの末尾が 2016 年 9 月 30 日になるまで繰り返し行う．

このような形式でデータセットを分割する方法は文献 [13] を参考にしたものである．ただし，当該文献のモデルはデータセットごとにモデルをリセットした形で用いるものであり，本稿で行う事前学習形式のものに比べ学習の収束が遅いという問題点がある．

4.3　実験設定

本稿の実験ではデータセット間の時間的な変化を考慮するために，遅延数を 4 とした回帰予測を行う．例えば，最初の時点では 1 日目から 4 日目までの説明変数を用いて学習を行い，5 日目の目的変数を予測する形になる．概略図を図 5 に示す．

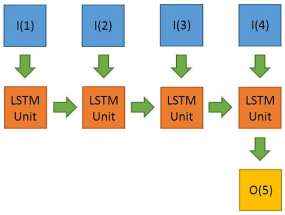

図 5　Diagram of LSTM for financial time series [13].

前節の処理で得られたデータセットを用いて，LSTM ないしは深層 LSTM を学習モデルとして学習を行い，回帰分析を行う．どちらの LSTM になるかは前節で述べた LSTM の層の数に関する超パラメータによって決定される．ただし，層の深い LSTM においては事前学習を行わない場合に浅い層でのみ学習が行われ，深い層の箇所ではほとんど学習が行われないという問題が存在する．そこで，DBN を用いた事前学習を株価指数のデータの初日に対して行い，結果から得られた重み及びバイアスでモデルのサンプリングを行うことで上記の問題の解決を試みた．

また，本実験においては超パラメータ探索の試行回数だけ LSTM の学習を行う必要があり，膨大な時間がかかることが予測される．このため，我々は 1 組のデータセットで学習を行う際の制限時間を 1 分とし，1 分後には次の組のデータセットに移行するように実験を設定した．

評価指標として，本稿では検証データおよびテストデータに対する平均絶対パーセント誤差（MAPE）を使用した．これは予測値と真値の間の誤差を割合として表現するものであり，値が小さいほど予測値と真値の間の誤差が少ないことを意味する．式は以下のようになる．

$$\frac{100}{n}\sum_{k=1}^{n}\left|\frac{f_i - y_i}{y_i}\right| \tag{18}$$

この指標は尺度が異なるデータセット群に対しても同一の評価尺度で評価可能な指標であり，通貨価値の違いにより株価の価格帯が異なるような本稿の実験においては適切な指標であるといえる．

本稿においては，LSTM モデルの誤差関数として MAPE を採用している．また，各最適化において評価に使用する出力値は各データセットの組ごとに取得された MAPE の値の平均値を採用している．

実験対象とした手法は，開始年順に行うマルチタスク・ベイズ的最適化，ランダムな順で行うマルチタスク・ベイズ的最適化，従来のベイズ的最適化，ランダム探索法の 4 つである．次節で実験結果を示す．

4.4　実験結果

4.4.1　検証データに対するマルチタスクベイズ的最適化の有効性の調査

はじめに，我々は前節で示した手法のうち以下の手法に対して検証データの精度を比較する実験を行った．乱数の値による実験の依存性を排除するために，同一の実験を 3 回行った．比

較対象とした手法は，開始年順に行うマルチタスク・ベイズ的最適化，従来のベイズ的最適化，ランダム探索法の 3 つである．3 回の実験結果から得られた MAPE の値を平均化したものを表 2 に示す．

表 2　Optimization results for all indexes in validation data.

	Multi-task BO	Single-task BO	Random Search
S&P 500	**12.66**	12.85	13.09
Nikkei 225	20.43	**18.52**	24.40
Hang seng	11.88	**11.72**	19.21
CSI 300	16.19	**15.99**	17.24
Nifty 50	**14.73**	14.74	15.80

ベイズ的最適化を使用した手法はランダム探索に比べ優位性があることが表 2 よりわかる．ベイズ的最適化により，超パラメータ探索の良し悪しを適切に評価する事が可能になったことが上記の実験結果に繋がったと考えられる．一方で，マルチタスク化したことによる優位性はそれほど見られない結果となった．特に Nikkei 225 指数に関してはマルチタスク・ベイズ的最適化は通常のベイズ最適化に比べて約 2%性能が劣る結果となった．我々は各実験ごとの結果においても同様の傾向が見られるのかを調査するために，各評価指数において最も性能が良かった超パラメータ値の組およびその設定が得られた際の評価値の値について調査を行った．

本実験において得られた結果の内，各評価指数において最も性能が良かった超パラメータ値の組について表 3 に示す．

表 3　Parameter settings.

	S&P 500	Nikkei 225	Hang seng	CSI 300	Nifty 50
U	13	26	62	110	13
La	2	1	6	1	0
Le	0.0519	0.0808	0.0349	0.0950	0.0441
B	72	44	83	46	43
E	4082	2471	8599	2167	7478
ρ	0.8383	0.8860	1.749	0.5351	0.5257
ϵ	1.90E-08	9.45E-8	7.27E-8	4.66E-8	1.80E-8
D	0.8243	0.4704	0.8514	0.7292	0.5581
MAPE	12.14	16.13	10.22	15.47	13.26
A	BO	MTBO	BO	MTBO	BO

ここで U はユニット数，La は層の数，Le は学習率，B はバッチサイズ，E はエポック数，D はドロップアウト率，そして A は最適化手法を表現している．

Nikkei 225 指数および CSI 300 指数においては，マルチタスク・ベイズ的最適化を用いた結果が最も性能が良く，一方で Hang seng 指数および Nifty 50 指数においては従来のベイズ的最適化が最も良い結果を示した．各超パラメータの値の組においては，学習率は $0.3\tilde{0}.9$ 付近に設定することにより性能が向上する可能性を持つことが示唆される結果となった．また，Hang Seng 指標以外の指標においては ρ は 1 よりも小さい値にし，層の数は少なめに設定すると良い結果が得られる傾向にあることがわかった．

このような結果となる原因を調査したところ，S&P 500 指数と Nikkei 225 指数の持つ銘柄の業種構成（これを A 群とする），および Hang Seng 指数と CSI 300 指数の持つ銘柄の業種構成（これを B 群とする）にはそれぞれ類似性が見られた．A 群は，業種の偏りが比較的少ないが，情報技術産業が比較的高い構成比を示している一方で，B 群は業種構成が金融業に偏っており，全銘柄の約半数を占めていることが調査により判明した．A 群のような業種構成は比較的先進国に見受けられるものであり，B 群のような業種構成は発展途上国の典型的な特性であることが知られている．

以上の調査および実験結果から，マルチタスク・ベイズ的最適化を用いた手法は業種構成の違いを捉えたパラメータ最適化を行っている可能性があると示唆される．

4.4.2　テストデータに対するマルチタスクベイズ的最適化の有効性の調査および予測評価

次に，我々は前節で示した手法のうち以下の手法に対してテストデータの精度を比較する実験を行った．比較対象とした手法は，開始年順に行うマルチタスク・ベイズ的最適化，従来のベイズ的最適化，ランダム探索法の 3 つである．比較対象とした手法は以下の 3 つの手法である．

前節の実験と同様に，乱数の値による実験の依存性を排除するために，同一の実験を 3 回行った．なお，マルチタスク化による性能の変化に着目するため，超パラメータの引き継ぎを行っていない指標である S&P 500 を分析対象から除外している．また，超パラメータの値の組によっては MAPE の値が発散してしまう傾向にあった為，そのような結果による評価の変動を避けるため中央値を取り実験評価を行った．3 回の実験結果から得られた MAPE の値の平均値，中央値および標本標準偏差を求めた結果を表 4 に示す．

表 4　Optimization results for all indexes in test data.

	MTBO			BO			RS		
	avg.	med.	s.d	avg.	med.	s.d	avg.	med.	s.d
NK	**18.68**	**16.34**	4.259	19.15	18.66	1.072	24.22	22.57	2.846
HS	**11.13**	**7.980**	7.419	12.72	10.52	3.816	17.16	8.390	7.213
CS	18.13	15.76	7.457	**16.73**	16.08	1.310	18.33	**15.73**	0.9780
NF	**14.41**	**11.05**	2.410	14.97	11.10	2.920	15.13	11.11	2.974

ここで，NK は Nikkei 225 指数，HS は Hang Seng 指数，CS は CSI 300 指数，NF は Nifty 50 指数を指す．マルチタスク・ベイズ的最適化を用いたモデルが CSI 300 指数を除いて最も良い性能を示しており，CSI 300 指数においても，偶然の結果で良い結果が得られたランダム探索に迫る結果が得られた．この結果はマルチタスク・ベイズ的最適化のテストデータに対する優位性を明確に示す結果であるといえる．

このような結果が得られる要因としては，ベイズ的最適化の「探索」と「活用」が関与していると考えられる.「探索」とは，ある超パラメータ探索において良い結果が得られた際に，更に良い結果を得られる可能性がある超パラメータの値の組を大きく変えることで最適値を探すことを指している．一方で「活用」とは同様の事象が発生した際に，良い結果が得られた超パラメータの値の組の近傍に更に良い結果が存在すると考えて値を小刻みに変化させながら最適値を探すことを指している．ベイズ的最適化の枠組みにおいてはこの「探索」と「活用」のトレードオフをどのように設定するかによって最適化の性質が変化する．本稿におけるマルチタスク・ベイズ的最適化は比較的

「探索」に重きをおいた探索をしていることが表 4 にある標本標準偏差（s.d）から推察され，結果としてこのような探索がにおける問題設定においては有効であったことが示唆される．

また，実際の株価予測において，各最適化手法における相違を調査するために，テストデータにおいて MAPE が良い値を示した超パラメータの値の組に対して，株価の予測を行った．このうち，Nifty 50 指数における 1 回目の実験で予測された結果を図 6 に示す．

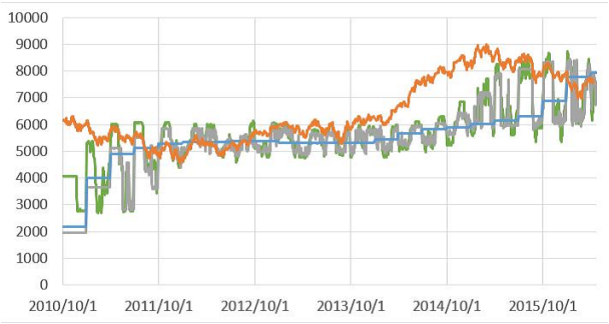

図 6　Prediction result of all optimization ways in Nifty 50.

横軸が日時，縦軸が株価である．また，オレンジの凡例が真値，緑がマルチタスク・ベイズ的最適化，灰色が従来のベイズ的最適化，そして青がランダム探索である．

全体として，1 次遅れのような株価予測となっている．個別に見るとランダム探索は予測値の変動が少ない一方で，ベイズ的最適化を用いたモデルは小刻みに予測値が変化している．また，マルチタスク・ベイズ最適化を用いたモデルは早い段階で真値に近づいており，データセット間の関連性を含んだ最適化が予測モデルの精度向上につながっていると考えられる．

4.4.3　データセットの順序の違いによるマルチタスク・ベイズ的最適化の性能変化

最後に，我々はデータセットの順序による実験結果への影響を考察する実験を行った．順序の影響を評価する指標は Nifty 50 指標とし，適用順序は最後に行われるようにした．それ以外の指標に関しては，すべての実験において適用順序が異なる形で行われるようにした．最適化を行ったデータの順序を実験回数とともに以下に示す．

- 1 回目:Hang Seng → CSI 300 → Nikkei 225 → S&P 500
- 2 回目:CSI 300 → S&P 500 → Hang Seng → Nikkei 225
- 3 回目:CSI 300 → Nikkei 225 → Hang Seng → S&P 500

上記の条件で実験を行い，各検証データセットごとの MAPE 値を比較したものが表 5 にある．

表 5　Multi-task Bayesian Optimization results for random order of indexes.

pattern	1st	2nd	3rd
1	HS: 22.78	CS 15.79	CS 15.60
2	CS: 17.04	SP 12.49	NK 19.92
3	NK: 29.11	HS 9.573	HS 14.08
4	SP: 12.94	NK 16.63	SP 12.52
5	NF: 13.66	NF 13.75	NF 14.41

ここで，SP は S&P 500 指数を意味する．CSI 300 指数から S&P 500 指数の最適化を行う順序にすることで，Nikkei 225 指標の性能が向上することから，前節の考察の妥当性を裏付ける結果となっていると予想される．本実験は更に実験回数を増やすことでより正確な傾向が得られることが期待される．

5.　おわりに

本稿では株価指標の予測を行う問題に対し，再帰型ニューラルネットワークの 1 つである LSTM を用いて解く際の超パラメータ探索について考察した．我々は関連するデータセットの最適化結果から得られる超パラメータの値の組を，次のデータセットのでベイズ的最適化におけるサンプルとするマルチタスク・ベイズ的最適化を提案した．主にテストデータにおいて業種構成が類似している株価指数の間では提案手法が有効であることを示し，および超パラメータの値の組についての一定の指針を示唆した．

本稿における今後の課題としては，データセットの順序対について考察することが挙げられる．また，文献 [1] で言及されているようなマルチタスク・ガウス過程を用いた手法の検討や LSTM 以外のモデルに対して実験を行うことも今後の課題である．

謝　　辞

本研究の一部は科学研究費補助金基盤研究 (B) (15H02703) の援助による．

文　　献

[1] Kevin Swersky, Jasper Snoek, Ryan P. Adams ”Multi-Task Bayesian Optimization” *Advances in Neural Information Processing Systems 26*(2013)
[2] Matthias Feurer, Jost Tobias Springenberg, Frank Hutter ”Initializing Bayesian Hyperparameter Optimization via Meta-Learning” *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*(2015)
[3] Hochreiter, Sepp and Schmidhuber, Jürgen ”Long Short-Term Memory” *Neural Comput.*(1997)
[4] Donald R. Jones ”A taxonomy of global optimization methods based on response surfaces” *Journal of Global Optimization*(2001)
[5] Alex Krizhevsky ”Learning multiple layers of features from tiny images” *Technical report, Department of Computer Science, University of Toronto,*(2009)
[6] 沖本竜義 経済・ファイナンスデータの計量時系列分析 朝倉書店 (2010)
[7] Ian Goodfellow, Yoshua Bengio, Aaron Courville ”Deep Learning”, chapter 20 *MIT Press*(2016)
[8] Hinton, Geoffrey E. Training Products of Experts by Minimizing Contrastive Divergence *Neural Comput.*(2002)
[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville ”Deep Learning”, chapter 10 *MIT Press*(2016)
[10] Jasper Snoek; Hugo Larochelle; Ryan P. Adams ”Practical Bayesian Optimization of Machine Learning Algorithms” *Advances in Neural Information Processing Systems 25*(2012)
[11] Carl E. Rasmussen and Christopher Williams ”Gaussian Processes for Machine Learning” *MIT Press,*(2006)
[12] T Tieleman, G Hinton ”Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude” *COURSERA: Neural networks for machine learning*(2012)
[13] Bao W, Yue J, Rao Y ”A deep learning framework for financial time series using stacked autoencoders and long-short term memory.” *PLoS ONE 12(7): e0180944.*(2017)