

SCWを用いた株価変動予測

Predicting Stock Fluctuations using Soft Confidence-Weighted Learning

福田 ムフタル^{1*} Muhtar Fukuda¹

¹ 名古屋産業大学環境情報ビジネス学部

¹ Faculty of Environmental and Information Studies, Nagoya Sangyo University

Abstract: Stock price prediction is a long-time and challenging topic in financial forecasts. Although stock markets are affected by many uncertain factors, numerous effectual approaches have been proposed to predict financial market trends using machine learning algorithms, such as Support Vector Machine (SVM) and Deep Belief Network (DBN). In this research, we propose a new approach to predict short-term stock fluctuations using Soft Confidence-Weighted (SCW) Learning. The proposed method not only predicts stock trends, but also gives a quantitative measure for the stock fluctuations. We consider RoC time series of a related stock class as inputs, fluctuate up and down of a target stock as outputs, to train the prediction model with SCW. Some experimental results show that the approach is useful for practical purposes.

1 はじめに

株式は多くの投資家にとって魅力的な投資対象であり、企業にとっては資金調達の一手段である。そのことから常に需要と供給が発生し、そのバランスによって市場価格が形成され、またはその価格で取引されると考えられており、経済学の理論や豊富な取引経験を持ったプロの投資家たちが様々な方法で分析を行いながら将来の経済動向を予測し、投資を行ってきた。

しかし、IT技術を基盤とした株式投資環境の発展、魅力的な投資商品の拡充、個人投資家を含めた市場参加者の拡大、マーケットのグローバル化、更にはSNS等を手段としたソーシャルメディアネットワークの普及により、株価形成に影響を与える要素が複雑化し、市場動向を伝統的な経済学や金融工学の理論だけでは説明できないケースが増え、投資パフォーマンスの予測が一層困難になってきた。そのような状況から、経験の浅い個人投資家に限らず、プロの機関投資家にとっても利益の最大化とリスクの最小化に役立つようなツールや方法の必要性が高まってきた。

また、人工知能の分野では前述の状況と並行して Deep Learning (DL)[1] に代表される先進的な機械学習モデルが考案され、大規模なニューラルネットワークを構築し、様々なビッグデータに対して機械学習を行い、全パラメータを最適化するというパラダイムシフトが起

きた。DLの金融データへの応用例として Deep Belief Network(DBN)[8]を用いた為替レート変動の予測 [2] や上場企業倒産の予測 [3] 等の研究が報告され、その有効性が示されている。また、もう一つの有力な機械学習モデルである Support Vector Machine (SVM)[5]を用いた株価予測に関する研究も数多く発表され [6, 7]、高精度のトレンド予測を実現したものもある。SVMはカーネルトリック [4] による非線形の識別関数を構成できるように拡張された学習モデルであり、現在知られている多くの手法の中でも最も認識性能の高いモデルの一つとされている。

しかし、多階層のネットワーク構造である DBN[8]を用いて金融時系列データの学習と予測を行う場合、その学習（訓練）に大変な時間がかかる。金融時系列データには時間的従属関係があり、訓練期間を常に最新のものに更新し、再学習を繰り返す必要がある。この問題に関しては自己組織化マップ (SOM) を用いて訓練期間全体から部分的なデータを取り出し、その部分的なデータのみで学習を行うことで学習の効率化と学習時間の低減を試みた研究 [9] があり、一定の改善があったと考えられるが、分単位の時系列データで学習と予測を行うことを考えた場合、更に学習時間の短縮が必要である。

一方、SVM[5]を用いて金融時系列データの学習と予測を行う場合、その正解率が訓練データのサイズと属性値 (feature) の選択に左右される場合が大きい [7]。また、金融時系列データは非線形であり、SVMモデルの予測正解率を上げるためには、非線形カーネルを使用する必要がある。非線形カーネルには複数のパラメー

*連絡先：名古屋産業大学環境情報ビジネス学部
〒488-8711 愛知県尾張旭市新居町の田 3255-5
E-mail: fukuda@nagoya-su.ac.jp

タがあり、SVMの振る舞いによりかなり影響を及ぼすため、その調整を慎重にかつ時間をかけて行わなければならない。訓練期間を常に更新する必要性のある機械学習の場合には問題になる。

更に、以上の各モデルを含め、従来の予測モデルの殆どは一日後の株価の方向性一致率に重点をおいており、ある銘柄の次の日の株価を「上がる」と予測し、実際の株価が例えば0.01%でも上がった場合、それが正解とされる。それは論理的には正解となっても、実際の投資目的には不十分である。株価予測の場合、株価が上がるか下がるかだけではなく、その騰落の程度を定量的に示すことがより重要である。買って売ってもコストに見合ったリターンが望めない場合は何もせず、一定以上のリターンが望めるタイミングで売買を行うのが実用的である。

本研究では、オンライン学習の一つである Soft Confidence-Weighted Learning (SCW)[10] を利用し、株価が上がるか下がるかの方向性だけではなく、株式の売買タイミングにも重点をおいた株価の値動きを予測する手法を提案する。

2 基本的な考え方

本研究では、以下の6つの考え方に基づいた株価予測を行う。

1) 予測できる可能性が高く、リスクが低い銘柄だけを予測の対象にする。今回は、予測対象として日経平均株価指数 [11] を考える。個別の銘柄の値動きはいろいろな要素の影響を受ける。次の日に上がると予測して株を購入し、次の市場が始まる前に悪材料が出た場合、予測が大きく外れることになる。格付け会社によってレーティングが変更されたり、目標株価がわずかに変更されただけで株価が動き、関連銘柄にも影響が出る場合がある。その点で日経平均株価指数は比較的に安定しており、予測しやすい。また、日経平均株価指数に一对一に連動するETFが幾つかあり、流動性が非常に高く、いつでもどこでも売買できる。

2) 一つの銘柄の株価は独立に動くことは殆どなく、複数の銘柄と直接または間接的に影響しあって相関的に動くと考え。従って、一つの銘柄の値動きを予測するためには関連性の高い複数の銘柄の時系列データの組み合わせを入力データとし、その関連銘柄群を一定の手順（後述）に基づいて常に機械的に選択し直す。その銘柄群を「ベース銘柄集」と呼ぶことにする。

周知の通り、株価の値動きは、需給、株価材料、投資家心理、為替、経済政策、国内外の政治経済状況、地政学、影響力の高い人物の発言、マスメディアなど様々な要因の影響を受ける。しかし、セクション1で述べたように、現在のマーケットを取り巻く環境は劇的に変化し

ており、何かが発生したら瞬時に株価の値動きに反映されるし、一つの銘柄の値動きに影響を及ぼす銘柄群も部分的に入れ替わっているはずである。

3) 学習期間 (図1の(c)) として予測ポイント (図1の(b)) から数日間離れた学習起点 (図1の(a)) より過去10年間前後の株価時系列データを使用するが、常に最新のものになるように学習起点をずらし続ける。ベース銘柄集と学習期間を常に更新し続けることにより、他の全ての株価変動要因が株価時系列データに反映されている状態が維持される。

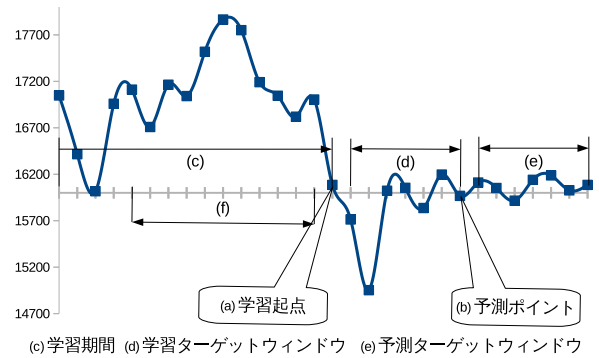


図1: ウィンドウ・学習起点・予測ポイント

4) 本研究では、次の日の株価の値動きではなく、次の日から数日間のウィンドウ内の値動きを学習や予測のターゲットにする。このウィンドウをターゲットウィンドウと呼び、学習ターゲットウィンドウ (図1の(d)) と予測ターゲットウィンドウ (図1の(e)) の二つに分けて考える。日数がターゲットウィンドウのサイズになる。ここでもやはり予測しやすいターゲットウィンドウサイズを考える。

5) 入力データに関してある時点から見た過去一定期間内の株価のトリプルボトムとトリプルトップの変動率を入力データの属性値 (feature) とし、その期間を属性ウィンドウ (図1の(f)) と呼ぶ。そして、予測対象銘柄の値動きと相関性の高い複数銘柄の属性値の組み合わせを一次的な入力データとし、学習期間の入力データを行に持つ行列 X^0 を作り、

$$X^0 = U\Sigma V^t \quad (1)$$

のように特異値分解し、行列 X^0V の各行を最終的な入力とする。予測ポイント t_0 の予測を行う時の一次的な入力データ $x_{t_0}^0$ に関して、 $x_{t_0}^0 = x_{t_0}^0 V$ を最終的な入力とする。

6) 本研究では、売買タイミングの割り出しにも重点をおいた予測を行う。騰落の程度を定量的に測る指標として「騰落強度」を導入する。「騰落強度」が大きければ大きいほど、売買タイミングに近付いたと判断する。

SCWを含めたオンライン学習では、入力 $x_t \in \mathcal{R}^d$ と機械学習によって更新される重み $w \in \mathcal{R}^d$ に対して、

$$\hat{y}_t = \text{sgn}(x_t \cdot w), \hat{y}_t \in \{+1, -1\} \quad (2)$$

でもって入力 x_t が属するクラスを予測する。

本研究では、全ての入力が d 次元空間の点であり、それらの点が法線ベクトル w を持つ超平面によって二つに分離され、点と分離超平面との距離が大きければ大きいほどそのクラスへの属性が強いと見做し、その距離(符号付き)を「騰落強度」とする。即ち、騰落強度 FS (Fluctuation Strength) を次のように定義する。

$$FS = \frac{x_t \cdot w}{|w|} \quad (3)$$

ここで、 $|FS|$ は騰落の強さ、 $\text{sgn}(FS)$ は上がるか下がるかの方向になる。しかし、騰落強度が最も大きい場合変動率が一番大きくなるという訳ではなく、その傾向が強いことを表す量になる。本研究では、その妥当性を実験によって検証する。

3 オンライン学習と SCW

オンライン学習は機械学習モデルにおける学習アルゴリズムの1つであり、入力データを1つずつ読み込んでモデルの更新を繰り返すことで学習を行い、時系列データの学習によく利用されている。オンライン学習では、入力の時系列データ $x_t \in \mathcal{R}^d$, $t = 1, 2, \dots, T$ に対して、そのラベル $\hat{y}_t \in \{+1, -1\}$ を予測し、正解ラベル $y_t \in \{+1, -1\}$ との差(損失)を損失関数 $l((x_t, y_t), \hat{y}_t)$ で計算し、その損失が一定の基準によって学習モデルの重みの更新に利用され、累積損失 $\sum_{t=1}^T l((x_t, y_t), \hat{y}_t)$ を最小化することで、学習が終了する。

オンライン機械学習には、他にも State-of-the-art の学習モデルとして CW[13], AROW[14], NAROW[15] などがあり、超空間上の点と平面の距離を騰落強度とする本研究の目的には合致するが、次の五つの理由から SCW[10] を利用することにした。1) 他のアルゴリズムの長所を持ちつつ、ノイズや直前のデータへの脆弱性が緩和されている。本研究では、直前のデータで学習を行うし、直後のデータの予測も行う。2) 他のアルゴリズムと同様 SCW にもいくつかのパラメータがあるが、パラメータを適当に調整しても識別率に殆ど影響がない。3) 他のアルゴリズムよりも識別能力が高い。4) 線形分離不可の入力データにも対応可能である。5) 共分散行列の更新を高速に行うことが可能である。

SCW に関してはオリジナルの参考文献 [10] で詳しく述べられているので、ここではその概要だけを述べる。SCW では CW や AROW の CW 系と同様、重みベクトル w が平均ベクトル μ , 共分散行列 Σ のガウス分

布に従うと仮定するが、損失関数 $l^\phi(\mathcal{N}(\mu, \Sigma); (x_t, y_t))$ を独自に定義している。

$$l^\phi(\mathcal{N}(\mu, \Sigma); (x_t, y_t)) = \max(0, \phi \sqrt{x_t^\top \Sigma x_t} - y_t \mu \cdot x_t)$$

ここで、 η は $r_w \sim \mathcal{N}(\mu, \Sigma)$, $P[y_t(w \cdot x_t) \geq 0] \geq \eta$ を満たすパラメータ、 Φ^{-1} は逆累積分布関数、 $\phi = \Phi^{-1}(\eta)$ である。

```

1 パラメータ:  $C > 0, \eta > 0$ 
2 初期化:  $\mu_1 = (0, 0, \dots, 0), \Sigma_1 = I, \phi = \Phi^{-1}(\eta),$ 
    $\psi = 1 + \frac{\phi^2}{2}, \xi = 1 + \phi^2$ 
3 for  $t = 1, 2, \dots, T$  do
4    $x_t \in \mathcal{R}^d$  を受け取る
5   ラベル  $\hat{y}_t$  を予測する:  $\hat{y}_t = \text{sgn}(x_t \cdot \mu_t)$ 
6   真ラベル  $y_t \in \{+1, -1\}$  の提示と損失の計算:
    $l_t = \max(0, \phi \sqrt{x_t^\top \Sigma_t x_t} - y_t \hat{y}_t)$ 
7   if  $l_t > 0$  then
8      $\mu_t$  と  $\Sigma_t$  の更新を行う:
      $\mu_{t+1} = \mu_t + \alpha_t y_t \Sigma_t x_t,$ 
      $\Sigma_{t+1} = \Sigma_t - \beta_t \Sigma_t x_t^\top x_t \Sigma_t$ 
9   end
10 end
```

Algorithm 1: SCW の学習アルゴリズム

SCW の学習アルゴリズムにおける α_t と β_t は更新係数と呼ばれ、式 (4) と (5) によって計算される。

$$\alpha_t = \min\{C, \max\{0, \frac{1}{v_t \xi} (-m_t \psi + \sqrt{m_t^2 \frac{\phi^4}{4} + v_t \phi^2 \xi})\}\} \quad (4)$$

$$\beta_t = \frac{\alpha_t \phi}{\sqrt{u_t + v_t \alpha_t \phi}} \quad (5)$$

ここで、

$$u_t = \frac{1}{4} (-\alpha_t v_t \phi + \sqrt{\alpha_t^2 v_t^2 \phi^2 + 2v_t})^2$$

$$v_t = x_t^\top \Sigma_t x_t, m_t = y_t (\mu \cdot x_t)$$

SCW では、 α_t の計算式が2種類あり、その種類によって“SCW-I”, “SCW-II”と呼ばれており、本研究では前者のほうを用いる。

実際の学習では、Step 3~9 を複数回繰り返し、最終的な μ_t をモデルの重み w として使用する。パラメータ $C(> 0)$ と $\eta(> 0)$ の値は、取り扱う時系列データ、その属性値、学習期間、学習ウィンドウやターゲットウィンドウのサイズなどによって調整が必要であると思われるが、本研究では一度学習ウィンドウとターゲットウィンドウのサイズを固定し、 C に関して $\{2^{-4}, 2^{-3}, \dots, 2^3, 2^4\}$, η に関しては $\{0.50, 0.55, \dots, 0.90, 0.95\}$ でのクロスバリデーション実験を行って C と η の値を決定した後、学習ウィンドウとターゲットウィンドウのサイズを適当に変更して実験を行ったところ、パラメータ値をそれぞれ決定して学習させた場合と一つに固定して学習さ

せた場合とで、その認識率の差が1%未満、繰り返し回数も数パーセント未満に抑えられており、固定のパラメータ値を使用する。そうすることで、学習時間を大幅に抑えることができる。今回のクロスバリデーション実験で決定したパラメータ値は $C = 0.125$, $\eta = 0.7$ である。

SCW を利用したことで学習時間を大幅に低減できるメリットはもう一つある。SCW の学習アルゴリズムの損失関数関数 l_t の計算、及び μ_t と Σ_t の更新では、数千行 × 数千列に及ぶ共分散行列 Σ_t に関する行列計算が複数カ所あり、それにはかなりの計算時間が必要になる。しかし、CW 系の SCW では Σ_t の対角線以外の要素を常に 0 にしても認識率に大きな影響がないことが知られており [16]、本研究の実験でも確認しているので、ベクトルの内積計算だけで済ませることができる。

4 提案手法による予測実験

株価の値動き、及び売買タイミングの予測に関する基本的な考え方は2で述べた通りである。

このセクションでは、まずベース銘柄集の自動選択、入力データを構成する属性値の決定、ラベル付け基準値の決定、及び入出力データの作成について述べる。それから、日経平均株価指数を対象とした予測実験を行い、株価の値動き予測に関して SCW に識別能力があることと予測ポイントの見極めに有効であることを示す。

4.1 ベース銘柄集の自動選択

一つの銘柄、日経平均株価指数のような株価指数、あるいは特定の業種に連動するように設定された Exchange Traded Fund (ETF) などは独立に動くことは殆どなく、その市場価格が複数の銘柄と直接または間接的に影響しあって相関的に変動すると考えられる。構成銘柄リストに含まれていない場合でさえその可能性がある。従って、一つの銘柄の値動きの予測を行う場合、関連性の高い銘柄群の時系列データの組み合わせを入力データとするのが自然である。そのような銘柄郡をベース銘柄集と呼ぶ。

ベース銘柄集をどの株式市場の上場銘柄から選ぶのか、国内外から同時に選ぶのかなどを検討する必要があるが、今回は東京証券取引所で取引されている全ての銘柄をベース銘柄集の選択対象とする。

ベース銘柄集を次のような手順で選択する。まず便宜上、選択対象の全銘柄の集合を \mathcal{S} 、学習期間を T 、銘柄 $s \in \mathcal{S}$ の学習期間中の売買成立日数を T_s 、 s の最大連続売買不成立日数を N_s ($T = T_s + N_s$)、基準最低取引成立日数を T_0 、基準最大連続不成立日数を N_0 とする。

Step0 予測対象銘柄 $s_0 \in \mathcal{S}$ を一つ決定する。話を進めやすくするため、全期間中において売買不成立日のない銘柄を選ぶ。即ち、 $N_{s_0} = 0$ になるような銘柄を選ぶ。

Step1 $\mathcal{S}_1 = \{s \in \mathcal{S} \mid T_s \geq T_0, N_s \leq N_0\}$ とする。即ち、できるだけトータルで売買不成立日が少くかつ連続の売買不成立日の少ない銘柄を選ぶ。 $s \in \mathcal{S}_1$ に売買不成立がある場合、3次 Spline による補間を行う。

今回の実験で $T = 2000$, $T_0 = 1900$, $N_0 = 2$ としており、そうした場合、 $|\mathcal{S}_1|$ は 900 前後になる。

Step2 \mathcal{S}_1 から時価総額上位 $P_1\%$ の銘柄を選び、 \mathcal{S}_2 とする。 $P_1 = 80\%$ とした場合、 $|\mathcal{S}_2|$ は 720 前後になる。

Step3 \mathcal{S}_2 から平均変動率の高い上位 $P_2\%$ の銘柄を選び、 \mathcal{S}_3 とする。ある銘柄 $s \in \mathcal{S}_2$ の予測ポイントから t 日目の株価終値時系列を $p_t, t = 1, 2, \dots, T$ とした場合の s の平均変動率 $AvgRoC(s)$ を次のように定義する。

$$AvgRoC(s) = \left(\prod_{k=0}^K \left(\frac{1}{\left[\frac{T}{2^k}\right]} \sum_{t=1}^{\left[\frac{T}{2^k}\right]} \left| \frac{(p_{t-1} - p_t)}{p_t} \right| \right) \right)^{\frac{1}{K+1}}$$

ここで、 $K \geq 0, 2^{K+1} \leq T$ 。

予測ポイントからの全 T 期間の平均変動率も、予測起点から $\left[\frac{T}{2}\right]$ までの平均変動率も、 \dots , $\left[\frac{T}{2^k}\right]$ までの平均変動率もバランスよく高くなるように算術平均と幾何平均の組み合わせを基準に上位を選んでいく。例えば、 $T = 2000$, $K = 5$ の場合、予測ポイントからそれぞれ向こう 2000, 1000, 500, 250, 125, 62 日間の絶対変動率の算術平均を計算してから幾何平均を計算する。

$P_2 = 80\%$ とした場合、 $|\mathcal{S}_3|$ は 576 前後になる。

Step4 $\mathcal{S}_3 \cup \{s_0\}$ から s_0 の s に対する平均絶対ベータ値の高い上位 N 個の銘柄を選び、 s_0 のベース銘柄集 $\mathcal{S}(s_0)$ とする。前の各 Step で s_0 が落とされる可能性があるため、ここで s_0 がベース銘柄集に入るようにしておく。今回の実験で $|\mathcal{S}(s_0)| = N = 400$ とした。

s_0 とある銘柄 $s \in \mathcal{S}_3$ の学習期間+ L 日目の t 日目の終値時系列をそれぞれ $p_t^0, p_t^s, t = 1, 2, \dots, T, T+1, \dots, T+L$ とし、 s_0 の s に対する平均絶対ベータ値 $AvgBeta(s_0 | s)$ を次のように定義する。

$$a_t^0 = \frac{1}{L} \sum_{i=1}^L p_{t+i}^0, \quad a_t^s = \frac{1}{L} \sum_{i=1}^L p_{t+i}^s$$

ここで L は属性ウィンドウのサイズ、 a_t^0 と a_t^s 、 $t = 1, 2, \dots, T$ はそれぞれ L 日間の移動平均値である。

$$\alpha_k = \left\{ \frac{p_1^0 - a_1^0}{a_1^0}, \frac{p_2^0 - a_2^0}{a_2^0}, \dots, \frac{p_{\lfloor \frac{T}{2^k} \rfloor}^0 - a_{\lfloor \frac{T}{2^k} \rfloor}^0}{a_{\lfloor \frac{T}{2^k} \rfloor}^0} \right\}$$

$$\beta_k = \left\{ \frac{p_1^s - a_1^s}{a_1^s}, \frac{p_2^s - a_2^s}{a_2^s}, \dots, \frac{p_{\lfloor \frac{T}{2^k} \rfloor}^s - a_{\lfloor \frac{T}{2^k} \rfloor}^s}{a_{\lfloor \frac{T}{2^k} \rfloor}^s} \right\}$$

$$\text{AvgBeta}(s_0 | s) = \left(\prod_{k=0}^K \left| \frac{\alpha_k \cdot \beta_k}{\beta_k} \right| \right)^{\frac{1}{K+1}}$$

ここで、 $K \geq 0$ 、 $2^{K+1} \leq T$ 。

ここでもやはり予測ポイントからの各 $\lfloor \frac{T}{2^k} \rfloor$ 、 $k = 0, \dots, K$ までの平均絶対ベータ値がバランスよく高くなるように幾何平均を基準に上位を選んでいる。一般的には他の銘柄の日経平均に対するベータ値を計算するが、ここではその逆である。

以上の各ステップをまとめると、できるだけ売買不成立日が少く、時価総額が大きく、変動率が大きく、予測対象銘柄の変動への貢献度が高い銘柄をベース銘柄集に選ぶことになる。本研究では、各ステップの順番も重要である。

4.2 属性値の決定

機械学習の場合、何を入力データの構成要素にするのが重要ファクターの一つであり、学習率、及び識別率に大きな影響を及ぼす場合が多い。その構成要素として移動平均値とRSI (Relative Strength Index) の組み合わせを用いたり [6]、 n 日間の株価時系列データをそれぞれウィンドウ内と全学習期間内で正規化したデータの組み合わせを用いた [9] 研究がある。

本研究では、属性値としてある時点 t_0 から見た過去 L 日間の属性ウィンドウ (図1の(f)) とし、属性ウィンドウ内の株価終値の時点 t_0 の終値からのトリプルボトムとトリプルトップの変動率を入力データの属性値とする。 L は属性ウィンドウのサイズになる。

ある時点とその直前の属性ウィンドウ内の終値時系列を p_t 、 $t = 0, 1, 2, \dots, L = 3 \times 2 \times \text{skip}$ とした場合のトリプルボトムの変動率は次のように求める。

Step1 $\frac{p_t - p_0}{p_0}$ 、 $t = 1, 2, \dots, L$ を昇順にソートし、その結果を c_t 、 $t = 1, 2, \dots, L$ とする。

Step2 $t_1, t_2, t_3 \in \{1, 2, \dots, L\}$ 、 $t', t'' \in \{t_1, t_2, t_3\}$ 、 $|t' - t''| \geq \text{skip}$ になるように $\{c_{t_1}, c_{t_2}, c_{t_3}\}$ を選

ぶ。即ち、できるだけ下位3つを選ぶが、時系列的に隣ではなく、互いに skip 日以上離れた値を選ぶようにする。

Step3 $\{c_{t_1}, c_{t_2}, c_{t_3}\}$ を本の時系列順に戻し、トリプルボトムの変動率 $\{l_1, l_2, l_3\}$ を得る。

トリプルボトムと反対の方法でできるだけ上位3つを選ぶことで、トリプルトップの変動率 $\{h_1, h_2, h_3\}$ を求めることができる。

今回の実験は株価が上がるかどうかの予測に $\text{skip} = 3$ ($L = 18$)、株価が下がるかどうかの予測に $\text{skip} = 2$ ($L = 12$) として行う。

次に、予測対象銘柄 s_0 のベース銘柄集

$$\mathcal{S}(s_0) = \{s_1, s_2, \dots, s_N\}, N = |\mathcal{S}(s_0)|$$

において、 $s \in \mathcal{S}(s_0)$ の時点 t のトリプルボトムとトリプルトップの変動率を $\alpha_t^s = \{l_{t_1}^s, l_{t_2}^s, l_{t_3}^s, h_{t_1}^s, h_{t_2}^s, h_{t_3}^s\}$ とし、 $x_t^0 = \{\alpha_t^{s_1}, \alpha_t^{s_2}, \dots, \alpha_t^{s_N}\}$ 、 $t = 1, 2, \dots, T$ を一次的な入力データとする。一次的な入力データができた時点で、 x_t^0 を行に持つ行列 X^0 を作り、

$$X^0 = U \Sigma V^t$$

のように特異値分解し、行列 $X^0 V$ の各行を最終的な入力とする。また、予測を行う時の一次的な入力データ x_t^0 に関して、 $x_t = x_t^0 V$ を最終的な入力とする。

ここでは、一次的な入力データ x_t^0 で学習させるのがよいか、それとも特異値分解によって変換した後の x_t で学習させるのがよいかについて実験を行い、後者のほうが学習率でも識別率でも優れていることを確認済みであり、今回の実験でも後者を用いる。もちろん、特異値分解には時間がかかるが特異値分解後の学習時間が劇的に速く、一次的な入力データでの学習に時間がかかり、トータルで考えた場合、特異値分解+学習にかかる時間がそうではない場合の時間より数倍の範囲内で収まっている。

4.3 ラベル付け基準値の決定

入力データが決まったところで、次に問題になるのは教師データであるラベルを何を基準に決めるかということである。セクション2でも述べたように、次の日の一日の株価の値動きではなく、次の日から数日間のターゲットウィンドウ内の値動きを予測対象とする。ターゲットウィンドウのサイズを M とする。

従来の研究では、株価の値動きに関して0を基準にし、当日 t と前日 $t+1$ の株価の変動率 $RoC = \frac{p_t - p_{t+1}}{p_{t+1}}$ を計算し、 $RoC \geq 0$ の場合は「上がり」(+1)、そうではない場合は「下がり」(-1)にラベル付けして学習と予測を行うことが多い。しかし、本研究では「上が

り」と「下がり」の基準値を別々に決め、学習と予測も別々に行う。

「上がり」の基準値は次のように決める。

Step1 ターゲットウィンドウ内の終値のトップ $\frac{1}{3}$ の平均値の変動率 $TRoC(3)$ を計算する。
終値 $\{p_M, p_{M-1}, \dots, p_1\}$ からトップ $m(3)$ を選び、 $h_1, \dots, h_{m(3)}$ とし、

$$TRoC(3) = \left(\frac{1}{m(3)} \sum_{i=1}^{m(3)} h_i - p_{M+1} \right) / p_{M+1}$$

$$\text{ここで, } m(3) = \begin{cases} 1 & \text{if } \frac{M}{3} < 1 \\ \lceil \frac{M}{3} \rceil & \text{otherwise} \end{cases}$$

Step2 学習期間内の $t = 1, 2, \dots, T$ に対して、**Step1** の $TRoC(3)$ を計算して降順に並べた

$$r_1, r_2, \dots, r_{\lceil \frac{T}{3} \rceil}, \dots, r_T$$

の中の $UR = r_{\lceil \frac{T}{3} \rceil}$ を「上がり」の基準値とする。

「上がり」と反対の方法で、ターゲットウィンドウ内の終値のボトム $\frac{1}{3}$ の平均値の変動率 $BRoC(3)$ を学習期間の全ての t に対して計算し、それを昇順に並べたものから「下がり」の基準値 $LR = r_{\lfloor \frac{T}{3} \rfloor}$ を求めることができる。

便宜上、これから時点 t の $TRoC(3)$ を $TRoC(t)$ 、 $BRoC(3)$ を $BRoC(t)$ とし、関数のように使用する。 UR と LR もそれぞれ「上がり」と「下がり」の基準値として使用する。

次に、ラベル付けの基準値 UR と LR が決定されたところで、

$$y_t^U = \begin{cases} +1 & \text{if } TRoC(t) \geq UR \\ -1 & \text{otherwise} \end{cases}$$

$$y_t^D = \begin{cases} +1 & \text{if } BRoC(t) \leq LR \\ -1 & \text{otherwise} \end{cases}$$

$$t = 1, 2, \dots, T$$

によってそれぞれ「上がり」と「下がり」のラベル付けを行う。そして、

$$Y^U = \{y_1^U, y_2^U, \dots, y_T^U\}, \quad Y^D = \{y_1^D, y_2^D, \dots, y_T^D\}$$

$X = X^0V$ (セクション4.2) とし、 $\{X, Y^U\}$ と $\{X, Y^D\}$ でそれぞれ上がるかどうか、下がるかどうかの学習を行う。

4.4 日経平均株価指数の予測実験

本研究で実験を行うためにまず参考文献 [10] に基づいて、実用に近い「SCW 学習・予測システム」を実装した。セクション3で述べた通り、SCW モデルの構造が極めてシンプルであり、本体部分の実装だけ考えれば、一般的なプログラミング言語と既存の Library を利用すれば数十行で済む。むしろデータの入手と加工のためのプログラミングに手間がかかったぐらいである。

今回の実験に使用するデータとして、東京証券取引所に上場している銘柄のうち、2016年2月末の時点で時価総額50億円以上、2003年以前から上場している1000銘柄前後、及び日経平均株価指数の時系列データをYahoo!ファイナンス [12] から取得した。そして日経平均株価指数を予測対象にし、複数の実験を行った。

本研究の原則は、予測しやすい銘柄だけを予測し、「上がる」、あるいは「下がる」と予測できたタイミングでしか売買しない。従って、実験結果もこの原則に基づいて評価する。

実験の前提条件は以下の通りである。

学習期間: $T = 2000$

セクション4.1の K : $K = 5$

ベース銘柄数: $|S(\text{日経平均})| = 400$

ターゲットウィンドウサイズ: $M = 3$

予測テスト期間: 2015年9月1日～2016年2月29日

属性ウィンドウサイズ: 「上がる」の場合 $L = 18$,

「下がる」の場合 $L = 12$

また、上がるかどうかと下がるかどうかは別々に学習し、別々に予測した。つまり、「上がる」の場合、上がると予測したものの中で何割正解があったかだけを問い、そうでない場合は問わないし、「下がる」の場合、下がると予測したものの中で何割正解があったかだけを問い、そうでない場合は問わない。

4.4.1 SCW の識別能力に関する実験

一つの学習モデルの予測正解率が何%までなら識別能力があると判断されるかについて基準がないだろうが、理論的には51%以上なら識別能力があると考えられる。一方、学習モデルによって学習・予測の対象になるデータの種類や特徴が違えば、その識別能力が上下する場合も考えられる。そこでSCWのモデルを利用した場合、どこまで予測可能かについて2015年9月1日から2016年2月29日までの半年の日経平均株価指数を対象に「上がる」と「下がる」の両方の実験を行った。その結果を図2と図3に示す。

図2と図3では、x軸は「上がる」「下がる」と判断した日付、y軸は日付の次のターゲットウィンドウの株価変動率(%)を表す。そして「上がる」に関しては83%、

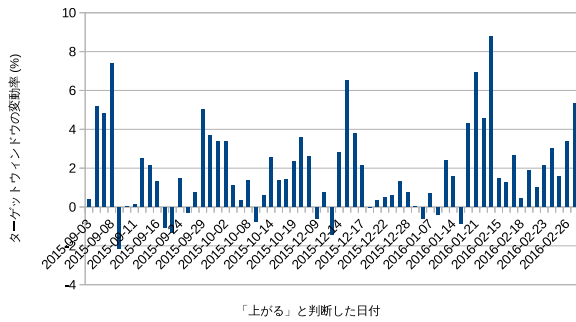


図 2: 「上がる」の場合

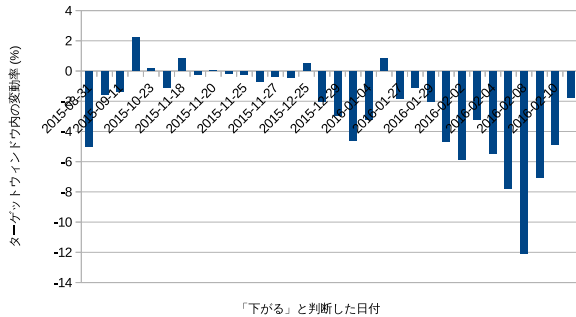


図 3: 「下がる」の場合

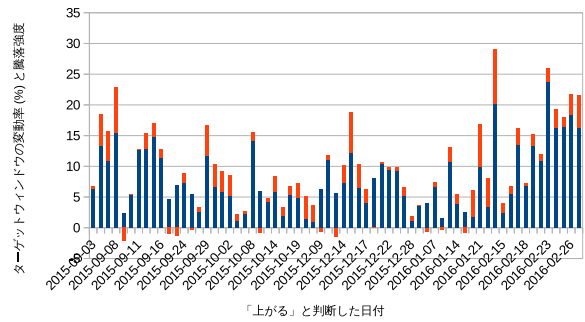


図 4: 「上がる」の場合

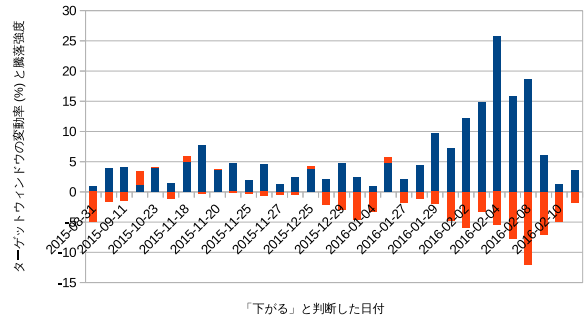


図 5: 「下がる」の場合

「下がる」に関しては 79%の正解率になった。この結果から、本研究の手法は有効であると判断できる。

4.4.2 売買タイミングに関する実験

本研究では、上がるかどうか、下がるかどうかを判断するだけでなく、上がるなら随分上がるか、下がるなら随分下がるかといった売買タイミングを見極めることにも重点をおいた予測を行う。その騰落の程度を定量的に測る指標としてセクション 2 の式 3 の通り、「騰落強度」を導入した。「騰落強度」が大きければ大きいほど、売買タイミングに近付いたと判断したい。

その目的のために、「上がる」「下がる」を判断した上に、騰落強度が大きければ、それに伴ってターゲットウィンドウの変動率も大きくなるかどうかについて調べた。売買タイミングに関しても「上がる」と「下がる」の両方の実験を行った。その結果を図 4 と図 5 に示す。

図 4 と図 5 では、x 軸は「上がる」「下がる」と判断した日付、y 軸は日付の次のターゲットウィンドウの株価変動率 (%) と騰落強度のスタックを表す。これらの結果から、「上がる」の場合も「下がる」の場合も騰落強度が大きければ大きいほど、変動率も大きくなる傾向にあることが分かる。「上がる」の場合は、平均値よ

り大きい騰落強度の場合、下がった銘柄が三つだけであり、「下がる」の場合も上がったのは三つであった。

また、図 4 と図 5 を併せて考えた場合、騰落強度が充分大きい時に株価が反対側になった場合はなく、そのようなタイミングが 10 回以上あり、半年でこれぐらいの回数のチャンスがあれば充分であると言える。このことから本研究の手法は有効であると判断できる。

5 結論

本研究では、オンライン学習の一つである Soft Confidence-Weighted Learning (SCW)[10] を利用し、株式の売買タイミングに重点をおいた株価の値動きを予測する手法を提案し、その有効性を示した。また、学習全体を「上がる」を主眼にした学習と「下がる」を主眼にした学習に分けて行った。株価の値動きに関して「上がる」「下がる」と予測されたものの中で実際にそうなった割合が 80%前後になった。

入力データを構成する属性値としてトリプルトップとトリプルボトムの変動率を組み合わせるという独自の方法を考案した。一つの銘柄の値動きを予測するためにその変動への貢献度の高い銘柄を動的に選択し、それらの属性値を組み合わせることも従来の研究ではあまり見られない。日経平均株価を構成する銘

柄が一年に一回入れ替えられており、似たようなところはあるが、今回提案した手法は機械的であり、主役銘柄が常に交代する現在のマーケットでは、今回提案した手法はより有効であると考えられる。

株式に関して、毎日その変動を予測して毎日売買しなければならないものではなく、「上がる」、「下がる」可能性が充分高くなった時点で売買を行い、一定以上の利益を確保する、あるいは損だけは出さないことは特に個人投資家にとって重要である。本研究では、その観点から、騰落強度を導入し、騰落強度が大きくなればなるほど変動率も大きくなる傾向にあることを確認した。

今後の課題として、ダウ平均株価指数、ナスダック平均株価指数等他の指数に関して同様の実験を検討していきたい。

謝辞

本研究を進めるにあたり、ご指導を頂いた恩師であり、名古屋大学名誉教授の稲垣康善先生に感謝致します。

参考文献

- [1] Hinton, G., Osindero, S. and Teh, Y. W.: A fast learning algorithm for deep belief nets, *Neural Computation*, Vol. 18, No. 7, pp. 1527–1554 (2006)
- [2] Chao, J., Shen, F. and Zhao, J.: Forecasting exchange rate with deep belief networks, *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1259–1266(2011)
- [3] Yeh, S., Wang, C. and Tsai, M.: Corporate Default Prediction via Deep Learning, In *The 34th International Symposium on Forecasting (ISF '14)*, (2014)
- [4] Aizerman, M., Braverman, E., and Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control 25*, pp. 821–837 (1964)
- [5] Boser, B. E., Guyon, I. M., and Vapnik, V. N.: A training algorithm for optimal margin classifiers, *Proceedings of the fifth annual workshop on Computational learning theory (COLT '92)*, pp. 144–152 (1992)
- [6] 淵井 亮, 鍾 寧: サポートベクターマシンを利用した株価の値動き予測, 電子情報通信学会技術研究報告, *KBSE2010-42*, pp. 43–48 (2011)
- [7] Shen, S., Jiang, H., and Zhang, T.: Stock market forecasting using machine learning algorithms, *CS229 (Machine Learning) at Stanford University*, accessed February 1, 2016, <http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearningAlgorithms.pdf> (2012)
- [8] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H.: Greedy Layer-Wise Training of Deep Networks, *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, accessed February 1, 2016, <http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf> (2007)
- [9] 小牧昇平, 白山晋: Deep Belief Network を用いた日経平均株価の予測に関する研究, 人工知能学会研究会資料, SIG-FIN-012-08, (2014)
- [10] Wang, J., Zhao, P., and Hoi, S. C. H.: Exact Soft Confidence-Weighted Learning, *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pp. 121–128 (2012)
- [11] 日経平均プロフィール, <http://indexes.nikkei.co.jp/nkave/>
- [12] Yahoo!ファイナンス, <http://finance.yahoo.co.jp/>
- [13] Dredze, M., Crammer, K., and Pereira, F.: Confidence-weighted linear classification, *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pp. 264–271 (2008)
- [14] Crammer, K., Kulesza, A., and Dredze, M.: Adaptive regularization of weight vectors, *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pp. 345–352 (2009)
- [15] Orabona, F. and Crammer, K.: New adaptive algorithms for online classification, *Advances in Neural Information Processing Systems 22 (NIPS 2010)*, pp. 1840–1848 (2010)
- [16] Ma, J., Kulesza, A., Dredze, M., Crammer, K., Saul, K.L., and Pereira, F.: Exploiting Feature Covariance in High-Dimensional Online Learning, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, accessed February 1, 2016, <http://jmlr.csail.mit.edu/proceedings/papers/v9/ma10a/ma10a.pdf> (2010)