深層学習と高頻度注文情報による株価動向推定

田代　大悟 [1*]　和泉　潔 [1]
Daigo Tashiro[1]　　Kiyoshi Izumi[1]

[1] 東京大学大学院工学系研究科
[1] Graduate School of Engineering, The University of Tokyo

Abstract:
In this paper, we propose order-based approach to predict future movements of a stock price. Our models employ a convolutional neural network(CNN) over embedded orders that have quantitative and qualitative variables. For each dataset of stock codes, the models outperform traditional feature-based approaches. Furthermore, we show that training under less influence of noise can be performed by applying an averaging filter to embedded feature space. Analysis of the embedding layer reveals that the models put emphasis on the features of market orders that are correlated with price return.

1　はじめに

アルゴリズム・トレード (以下アルゴ) とは，機関投資家などから注文を委託された証券会社が，小口化し売買を行う機械的取引方法である．アルゴはコンピュータを利用して自動的に発注と売買を高速に行う点で High Frequency Trading(HFT) と共通するが，HFT がマイクロ秒オーダーで高速かつ高頻度で小口売買で利鞘を稼ぐ一方，アルゴは比較的低頻度にかつ取引コストの低減することに主眼を置いて執行を行う．取引コストをいかに低減するかは，アルゴにおいて重大な課題であり，高度なアルゴリズム開発のために，多様な分野の学術研究や先端技術が応用されている [2]．なかでも，アルゴの主な執行戦略である VWAP (volume-weighted average price) 戦略 [1] と深層学習による価格予測を併用することにより，パフォーマンスを向上したとの金融機関からの報告があり，本研究でも VWAP の支援となるような価格予測モデルの構築を目指す．

前述の通り機械的な売買方法の台頭や金融市場の電子化と高速化に伴い，蓄積される注文データはサンプリング頻度が極めて高く，また膨大化している．これらは「高頻度データ」と呼ばれ，有効な利用が期待されている．市場価格は，トレーダーが出した注文を取引所が集計，約定処理の後に形成される．そのため，注文データには価格時系列データよりも多くの情報が含まれていると考えられる．しかし，この高頻度データを，タイムドリブンで執行を行うアルゴへの応用を考えた場合，不等間隔に観測される注文，取引をうまく扱う必要がある．

ここでは，そのような高頻度データに機械学習を応用した例を紹介する．Alec らは，板情報のある時点での仲値やビッドアスクスプレッドなどの特徴量を人手によって設計し，SVM(Support Vector Machine) による仲値の変動の予測を行っている [3]．また，特徴選択をしていない生の板情報の入力と，ニューラルネットワークを用いた複雑な関数近似による予測も行われている．Tsantekidis らは，板に累積した注文の価格と量を入力として，LSTM(Long Short-Term Memory) による予測を行い，SVM を大きく上回る結果を得ている [4]．

板情報だけを説明変数に用いる場合，時点々々でのアスクとビッドの強度や均衡を動的に辿ることができる．しかし，ベストアスクまたはベストビッドの量が減少した場合，それが成行注文によるものか，キャンセル注文によるものか判別がつかないというデメリットがある．成行注文とは一般的に即時約定する注文でトレーダーの強い意思を表したものである．さらに，成行注文とリターンに相関がある [5][6] ため，キャンセル注文のもつ意味，情報とは異なると考えられる．

そこで本稿では，前述の課題を克服するため，注文ベースでの株価の動向予測について説明する．2 章では，提案手法となる，注文を記号列に変換する前処理，CNN(Convolutional Neural-Network)，埋め込み特徴行列の平均化を用いた CNN について説明し，3,4 章で具体的な実験と結果考察を行う．

*連絡先：東京大学大学院工学系研究科システム創成学専攻和泉研究室，〒 113-8654 東京都文京区本郷 7-3-1，E-mail: m2016dtashiro@socsim.org

2 CNNによる短期価格動向予測

ここでは初めに，注文の記号化について説明する．2.2では，注文系列から，タイムドリブンに対応する特徴へ変換することを意識し，問題を設定する．最後に，CNNによる価格動向予測モデル，埋め込み特徴行列の平均化を用いたCNNによる価格動向予測モデルの構築を行う．

2.1 注文の符号化

注文には，価格，時刻といった数値情報 (量的変数) と，種類といったカテゴリ情報 (質的変数) を持つ．ここではこのような注文を記号に変換し，その時系列を記号列で表現することを目的とする．注文の特徴には，売成行 (Ask Market Order) ／買成行 (Bid Market Order) ／売指値 (Ask Limit Order) ／買指値 (Bid Limit Order) ／売キャンセル (Ask Cancel) ／買キャンセル (Bid Cancel) といった注文の種類の他，価格，注文量，時刻などがある．ここでは種類と価格，そして前の注文からの時刻差のみに限定する．まず，各指値注文と各キャンセル注文に対しては，注文の入った時点での板の最良売り気配と最良買い気配の平均を仲値として求め，価格と仲値との差の絶対値を保持する．時刻差は一つ前の注文との時刻差である．これは，注文の特徴として凡その発注者の情報を加えることを意図している．例えば，マイクロ秒オーダーでの注文は機械的なトレーダーによる注文だと識別できるようにしている．

次に，価格差と時間差をそれぞれ，ある分類規則に従いカテゴライズする．その後，各注文に識別符号を与える．成行注文2種に関しては，時刻差のみを考慮し識別符号を付与し，指値注文とキャンセル注文に関しては，仲値からの価格差と時刻差よって分類後，符号を付与する．

注文と識別符号間の変換は一意であり，符号から注文へは復元可能となっている．なお，記号に変換することで価格といった質的変数の情報を落とすことになるが，大量のデータの中からパターンを識別することで，タスクと注文の意味の関係を自ら獲得することを期待している．

2.2 問題設定

本節では，価格動向予測の問題設定を行う．まず不等間隔のタイムスタンプを持つ注文に対して，任意の一定間隔 T で区間を設ける．ここで，各区間内での注文の系列の集合を $\mathcal{C} = \{\mathcal{S}^1, \mathcal{S}^2, \ldots\}$ と表し，各系列 $\mathcal{S}^i$ は $\mathcal{S}^i = (x_1^i, x_2^i, \ldots)$ と書く．x_τ^i は，2.1で定義された，系列 $\mathcal{S}^i$ 内の τ 番目の注文を表現する記号であり，系列 $\mathcal{S}^i$ は，i 番目の区間における注文の可変長の記号列を表す．各注文 x_τ^i は，前処理にて分類される識別符号の集合で大きさ I の集合 $\mathcal{I}$ の要素である．

目的は，各系列 $\mathcal{S}^i = (x_1^i, x_2^i, \ldots)$ を説明変数として，$\mathcal{S}^i$ に対応する時間領域の終点から T' 後の価格の動向ラベル t^i の予測である．ニューラルネットワークで表現される関数のパラメータを θ とし，出力となる条件付き確率を $p(t^i|x_1^i, x_2^i, \ldots; \theta)$ で表す．教師あり学習によって，この条件付き確率が最大となる最適なパラメータ θ を探索する．

2.3 CNNを用いたモデル

モデルにはCNNを用いる．CNNは，入力の領域内での不変性を仮定し，畳み込みフィルタというパラメータを各窓において共有することによりパラメータの学習を容易にする．CNNは画像認識の分野だけでなく，自然言語処理の分野においても文書分類といったタスクで成功を収めている [7][8]．本研究でCNNを用いる理由として，可変長系列の学習に対してパディングを行うことで容易にミニバッチ学習を行える点，RNN(Recurrent Neural Network) であれば予測のタスクが系列の後方のデータに依存するのに対して，CNNは畳み込みの後に系列方向に最大プーリングを行うため，価格動向に相関が比較的小さいであろう指値注文やキャンセルが系列の後方に集中してもうまく学習することができるであると期待した点が挙げられる．

ここではCNNモデルの定式化を行う．まず，系列 $\mathcal{S}$ 内の注文 x_τ の埋め込みベクトル $\boldsymbol{x}_\tau$ を次のように得る．

$$\boldsymbol{x}_\tau = \boldsymbol{w}_{\text{embed}} \boldsymbol{x}_\tau^{\text{onehot}} \tag{1}$$

ここで，$\boldsymbol{w}_{\text{embed}} \in \mathbb{R}^{e \times I}$ は埋め込み行列．e は埋め込みベクトル $\boldsymbol{x}_\tau$ の次元数である．$\boldsymbol{x}_\tau^{\text{onehot}}$ は注文 x_τ を表すonehotベクトルである．系列 $\mathcal{S}$ はパディングによって長さ n で統一後，次のように表現する．

$$\boldsymbol{x}_{1:n} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \tag{2}$$

$\boldsymbol{x}_{i:i+j}$ は注文 $(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_{i+j})$ を連結したものを表す．窓幅 h に対して畳み込みを行うフィルタの重み行列を $\boldsymbol{w}_{\text{conv}} \in \mathbb{R}^{e \times h}$ とすると，次のような式を得る．

$$c_i = \tanh(\boldsymbol{w}_{\text{conv}} \cdot \boldsymbol{x}_{i:i+h} + b) \tag{3}$$

c_i は埋め込み後の注文の局所的な小行列 $\boldsymbol{x}_{i:i+h}$ を畳み込みこむことによって得られる新たな特徴．$\cdot$ はドット積，$b \in \mathbb{R}$ はバイアスである．ストライド幅1として，$(\boldsymbol{x}_{1:h}, \boldsymbol{x}_{2:h+1}, \ldots, \boldsymbol{x}_{n-h+1:n})$ に対してこの畳み込みを行うと，次のような新たな特徴ベクトルを得る．

$$\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}]^T \in \mathbb{R}^{n-h+1} \tag{4}$$

次に最大プーリングを行うことによって，一つのフィルタから得られる一つの特徴量 $\hat{c} = \max(\boldsymbol{c})$ を得る．k_{conv} 個のフィルタによる畳み込み演算と最大プーリングによって得られる特徴量 $\hat{\boldsymbol{c}} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{k_{\text{conv}}}]^T$ を全結合層の重み $\boldsymbol{w}_{\text{fully}} \in \mathbb{R}^{l \times k_{\text{conv}}}$ を用いて次のように変換する．

$$\boldsymbol{o} = \boldsymbol{w}_{\text{fully}} \hat{\boldsymbol{c}} \tag{5}$$

l は動向ラベルのサイズである．p_j を j 番目のラベルの確率とすると，得られる確率は以下のようになる．

$$p_j = \frac{\exp o_j}{\sum_{i=1}^{l} \exp o_i} \tag{6}$$

2.4 学習

(7) 式に示す目的関数を最小にするようパラメータ $\theta = (\boldsymbol{w}_{\text{embed}}, \boldsymbol{w}_{\text{conv}}, \boldsymbol{w}_{\text{fully}})$ を探索し，最適パラメータ $\theta = \theta^{\text{opt}}$ を得る．

$$L = -\sum_{i=1}^{C} \sum_{j=1}^{o} t_{i,j} \log p_j \tag{7}$$

ただし，$C = |\mathcal{C}|$ かつ $t_{i,j}$ は系列 $\mathcal{S}_i$ に対応するラベルを onehot ベクトルで表現した時の j 番目の要素である．

2.5 埋め込み行列の平均化を用いた CNN

金融市場での時系列データでは，一定の価格トレンドが観測されても，注文単位などのミクロ構造での明確なパターンは少ない．テキストデータとは異なり注文の時系列では，畳み込みフィルタが捉える局所的な領域での，注文の相互作用が小さいと考えられる．そこで，一定数の注文の平均をとることで，これらの影響を小さくすることを考える．平均化を行うのは，埋め込み行列の局所的な範囲を対象とする．

ニューラルネットワーク構造を Figure 1 に示す．2.3 での (2) 式に示す $\boldsymbol{x}_{1:n}$ に対して平均プーリングを，窓幅 $1 \times k_{\text{pool}}$，ストライド幅 $1 \times k_{\text{pool}}$ で適用する．プーリング前の注文時系列方向上下それぞれののパディングサイズ m_{pad} は以下のようになる．

$$m_{\text{pad}} = \begin{cases} 0 & (n \equiv 0 \pmod{k_{\text{pool}}}) \\ \lceil \frac{k_{\text{pool}} - (n \bmod k_{\text{pool}})}{2} \rceil & (\text{otherwise}) \end{cases}$$

ただし，$\lceil x \rceil$ は天井関数であり，x 以上の最小の整数を表す．床関数を $\lfloor x \rfloor$ とすると，プーリング後の特徴行列は，

$$\boldsymbol{x}_{\text{pool}} \in \mathbb{R}^{e \times \lfloor \frac{m_{\text{pad}} + n}{k_{\text{pool}}} \rfloor}$$

のように表すことができる．その後は 2.3，2.4 と同様に順伝搬および学習を行う．

3 実験

3.1 銘柄と期間

データは FLEX-FULL ヒストリカルデータを用いた．このデータは東証上場現物株式の各銘柄のティックデータを蓄積したものである．ティックデータには取引情報だけでなく，注文，取引の時刻，銘柄情報，価格，数量といった情報が含まれる．データは大和ハウス工業 (銘柄コード 1925)，積水ハウス (1928)，キリン (2503)，味の素 (2802)，東レ (3402)，旭化成 (3407)，三菱ケミカルホールディングス (4188)，第一三共 (4568)，JXTG ホールディングス (5020)，新日鐵住金 (5401)，東芝 (6502)，パナソニック (6752) の各々2013 年 7 月 1 日から 2014 年 6 月 30 日までの，全 12 銘柄，245 日のデータを用いた．

3.2 注文の記号列変換

2.1 で述べた手法で，注文の前処理を行う．本実験では，それぞれの特徴量に対して，以下のようなカテゴライズを行った．価格に関しては，注文の出現数の多い，仲値に近い注文ほど分類の粒度を細かく，出現数の少ない，仲値に遠い注文ほど粒度を粗く設定した．時刻差に関しては，500ms を境に機械的な注文か否か，さらに機械的な注文の中でも 20ms を境に高速な注文か比較的低速な注文なのか，識別できることを期待して設定した．このようにして得られる識別符号の集合 $\mathcal{I}$ の大きさは $I = 90$ となった．

$$\begin{pmatrix} \text{MarketOrder}^{\text{ask}} \\ \text{MarketOrder}^{\text{bid}} \end{pmatrix} \times \begin{pmatrix} \sim 20\text{ms} \\ 20\text{ms} \sim 500\text{ms} \\ 500\text{ms} \sim \end{pmatrix}$$

$$\begin{pmatrix} \text{LimitOrder}^{\text{ask}} \\ \text{LimitOrder}^{\text{bid}} \\ \text{Cancel}^{\text{ask}} \\ \text{Cancel}^{\text{bid}} \end{pmatrix} \times \begin{pmatrix} \sim 1 \\ 1 \sim 2 \\ 2 \sim 3 \\ 3 \sim 5 \\ 5 \sim 7 \\ 7 \sim 10 \\ 10 \sim \end{pmatrix} \times \begin{pmatrix} \sim 20\text{ms} \\ 20\text{ms} \sim 500\text{ms} \\ 500\text{ms} \sim \end{pmatrix}$$

種類　　価格　　時間差

例を挙げると，買成行注文の，前の注文から 30ms 後の注文であれば 4，売キャンセル注文のうち仲値から 2 の，前の注文から 5ms 後の注文であれば 75 の符号を与えている．

3.3 データセット

本節では，一つの銘柄において，1 日単位で連続する注文の記号列からデータセットを作成する手法を述べる．ある銘柄の 1 年分 245 日のうち，前 8 割の 196

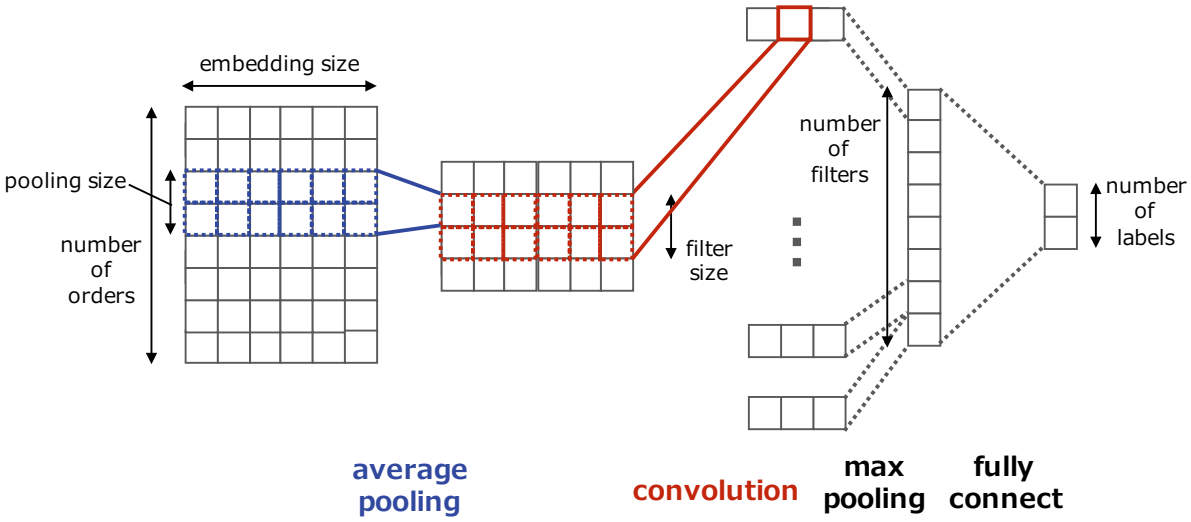

Figure 1: 埋め込み行列の平均化を用いた CNN モデルの構造.

日分を訓練用生データ，後 2 割の 49 日分を検証用生データとして分割する．訓練用，検証用それぞれに対して，以下の方法でデータセットを作成する (Figure2)．各日毎のザラ場内 (9:00〜11:30,12:30〜15:00) の注文に対して，9:00:00 を起点として 11:30:00 までの注文を 30 秒間隔で逐次サンプリングを行い，複数の可変長記号列を生成する．後場に対してもこれを同様に行う．i 日の系列集合のうち j 番目の注文記号列を $\mathcal{S}_i^j$ とする．$\mathcal{S}_i^j$ は区間 $[\tau^{j-1}, \tau^j)$ から得られたとする．ただし，$i \in (1, 2, \ldots, 245)$，かつ $j \in (1, 2, \ldots, 600)$ である．なお，j の属する集合のサイズ 600 はザラ場中 5 時間を 30 秒で割って求める 1 日の区間の数である．これで，訓練用 $196 \times 600 = 117600$，検証用 $49 \times 600 = 29400$ の系列数となる．訓練用に対しては，区画を決定する起点を 10 秒ずつずらして，これまでの処理を行う．つまり，9:00:10 と 9:00:20 を起点としてそれぞれ 30 秒間隔でサンプリングを行っていく．これは Data Augmentation のようなもので，注文の系列の一部に重複を認めるものの，訓練用のデータセット数を大きくするために行う．このようにして，訓練用の系列数を 3 倍とした．次に，ある $\mathcal{S}^j$ に対して，対応する区間の終点の株価 p_{τ^j} と終点から 30 秒後の株価 $p_{\tau^{j+1}}$ を比較して次のようにラベリングを行う．

$$t^j = \begin{cases} 0 & (p_{\tau^j} > p_{\tau^{j+1}}) \\ 1 & (p_{\tau^j} < p_{\tau^{j+1}}) \end{cases} \tag{8}$$

なお，価格に変動がない場合，つまり $p_{\tau^j} = p_{\tau^{j+1}}$ となった系列と，注文数が 20 以下の系列に関しては，本研究での予測の対象外としてデータセットから除外した．さらに，訓練用に関しては，正値ラベルと負値ラベルの系列数を同一とした．これらの処理をすべての銘柄に対して行い，得られる訓練用データセットは，最大で銘柄コード 6752 の 155202 件，最小で銘柄コード 4188 の 36364 件となった．

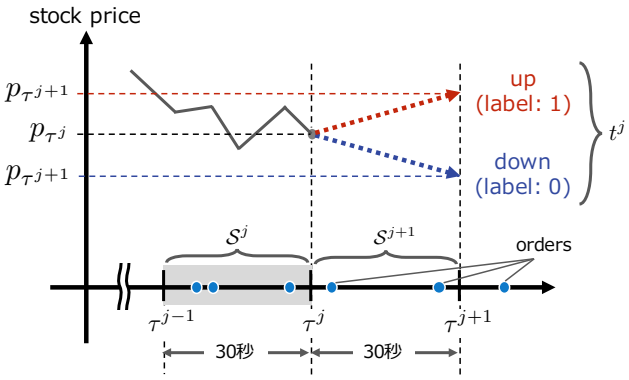

Figure 2: 系列 $\mathcal{S}^j$ のサンプリングとラベリング.

3.4 実装

各銘柄で用意したデータセットに対して，それぞれのモデルを用意する．(7) 式が最小となるように，最適化を行う．埋め込みサイズ $e = 5$，畳み込みフィルタの窓幅 $\boldsymbol{h} = \{1, 3, 5, 7\}$ とし，各窓幅に対して 5 つのフィルタ，全 20 フィルタを用意した．埋め込み行列の平均化を用いたモデルでは，平均プーリングの窓幅 $k_{\text{pool}} = 10$ とする．勾配降下法の最適化アルゴリズムには Adam を，$\lambda = 0.0005$ とした重みへの L2 正則化を，全結合層では比率 0.5 でのドロップアウトを使用し，バッチサイズ 100 としたミニバッチでの学習を 50 エポック行った．深層学習フレームワークとして chainer を使用し実装した．

3.5 ベースライン

ベースラインとして，ロジスティック回帰，非線形SVM，MLP(Multi Layer Perceptron) を用いた．提案手法と同様に，各銘柄に対してそれぞれのモデルを学習する．入力としては，系列 $\mathcal{S}$ 内の注文の頻度を各注文の種類 I の長さのベクトルで表現したものを用いる．可変長の系列 $\mathcal{S}$ の注文に対して，これらの手法では入力次元を一定にする必要があるからであり，このようなナイーブな手法をとることとする．MLP の中間層 2 層の次元はそれぞれ 8,4 とした．

4 結果と考察

4.1 モデルの評価

12 銘柄の各 5 手法，全 60 の学習済みモデルを，検証用データを用いて予測と評価を行う．3.3 で付与した，上昇したか下落したかというラベルに対してそれぞれ F 値を求め，サンプル数で加重平均をとったもので評価する．検証データのサンプル数は，最大で銘柄コード 6752 の 12835 件，最小で銘柄コード 4188 の 3122 件となった．結果を Table1 に示す．CNN では，銘柄コード 2503, 2802 の以外で，ベースラインすべてのモデルを上回るまたは等しい結果を残した．A-CNN モデルでは，すべての銘柄におけるベースラインおよび CNN モデルを上回る結果となった．CNN と A-CNN の提案手法と，ベースラインとを比較すると，ベースラインの入力が時系列情報を落としているため，提案手法が上回ったと考えられる．しかしそれは，可変長時系列をそのまま扱うことのできる提案手法の利点によるものである．

4.2 埋め込み層の分析

CNN と A-CNN の埋め込み層の分析を行った．各モデルの埋め込み行列 $\boldsymbol{w}_{\text{embed}} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{90}]$ の各列 $\boldsymbol{w}_i$ は識別符号 i の注文の埋め込みベクトルに対応する．これは各注文を表現したベクトルであり，そのノルムはニューラルネットの発火の強さであると考えることできる．Table 2 には，各モデルにおける埋め込みベクトルのノルムを注文の種類毎に集計平均したものを，さらに銘柄毎に平均をとったものである．CNN, A-CNN ともに，成行注文のノルムが，指値注文，キャンセル注文のそれに比べて大きい．さらに A-CNN のそれの方がより大きいノルムを持つ．これから，すべての注文の中から成行注文の特徴を強く捉えようと，モデルが学習していることがわかる．本研究でのタスクはリターンではなく価格動向であるが，成行注文とリターンに相関があるという主張 [5][6] を支持しているものだと考えられる．

4.3 成行注文比率と評価値

次に，なぜ銘柄間で評価値に差異があるのかを調査した．ベースラインで最もスコアの高かった MLP と CNN，A-CNN の 3 手法のそれぞれ 12 個のモデルの評価値と成行注文比率との関係を Figure 3 に示す．成行注文比率とは，検証データ内における，すべての注文に対する成行注文の割合である．それぞれのモデルに対して線形フィッティングを施すと，決定係数は MLP, CNN, A-CNN の順で高くなる．相関係数も CNN で 0.909(p 値 4.29×10^{-5})，A-CNN で 0.940(p 値 5.67×10^{-6}) と非常に高い．提案手法は，成行注文の密度が高ければ予測精度が高くなるということを示している．

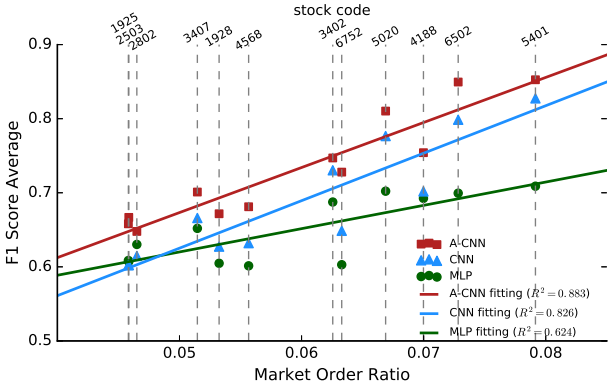

Figure 3: 各銘柄 × 評価値上位 3 手法のモデルにおける成行注文比率と F1-average との関係．R^2 は決定係数．

逆に成行注文以外の指値注文やキャンセルの比率が大きくなると精度が低くなることから，指値注文やキャンセルは価格動向との相関が低くノイズとなっていたと考えられる．CNN と A-CNN を比較すると，注文の意味空間で平均とることが，順伝搬，逆伝搬でのノイズを除去し，成行注文に反応するように訓練する一助になったと考えられる．

5 まとめ

本稿では，注文の符号化と，一定時間内でのサンプリングにより得られる可変長注文記号列を，CNN によって固定長の特徴に変換後，予測を行うモデルを提案した．分析の結果，価格変動と相関のある成行注文をとらえるような訓練をすることわかった．今回のタスクにおける精度の問題を改善するためには，成行注文のみを入力とし，シンプルな CNN や RNN を用いた場合の方が精度が高くなると考えられる．しかし，他の問

Table 1: 各手法 × 各銘柄のモデルの F1-Average による評価．A-CNN は 2.5 での埋め込み行列に平均化を適用したモデル．

手法	銘柄コード											
	1925	1928	2503	2802	3402	3407	4188	4568	5020	5401	6502	6752
Logistic	0.602	0.602	0.596	0.614	0.662	0.631	0.661	0.600	0.666	0.691	0.674	0.585
SVM	0.588	0.594	0.593	0.613	0.674	0.646	0.662	0.592	0.689	0.701	0.692	0.589
MLP	0.610	0.607	0.608	0.630	0.689	0.652	0.693	0.608	0.713	0.709	0.701	0.605
CNN	0.610	0.627	0.605	0.616	0.732	0.667	0.703	0.633	0.777	0.828	0.799	0.649
A-CNN	0.659	0.672	0.667	0.649	0.747	0.701	0.754	0.683	0.810	0.853	0.850	0.728

Table 2: 埋め込みベクトルのノルム平均．

注文の種類	手法	
	CNN	A-CNN
$\text{MarketOrder}^{\text{ask}}$	1.31	2.22
$\text{MarketOrder}^{\text{bid}}$	1.32	2.17
$\text{LimitOrder}^{\text{ask}}$	0.382	0.323
$\text{LimitOrder}^{\text{bid}}$	0.402	0.307
$\text{Cancel}^{\text{ask}}$	0.373	0.279
$\text{Cancel}^{\text{bid}}$	0.373	0.286

題設定下での注文の意味や特徴を探るには，本研究で行ったようなすべての注文を用い，平均化を行う手法は活用できると考えられる．今後，関連研究で行われているような，指値，キャンセルの相関が比較的大きいであろう仲値の予測を行い，埋め込みベクトルの関係がどのように変化するかを調査したいと考えている．また本研究では調査できていないが，予測の結果が単に価格のトレンドフォローであるか否かの検証，注文のその他の特徴である，価格や時間差に意味があるかなどの分析を今後検討している．

参考文献

[1] R. Almgren and J. Lorenz, "Adaptive Arrival Price," *Algorithmic Trading III, Institutional Investor*, pp. 59–66, 2007.

[2] 杉原慶彦 「取引コストの削減を巡る市場参加者の取組み：アルゴリズム取引と代替市場の活用」『金融研究』第 30 巻第 2 号，29–88 頁，2011 年．

[3] A. N. Kercheval and Y. Zhang, "Modelling high-frequency limit order book dynamics with support vector machines," *Quantitative Finance*, vol. 15, no. 8, pp. 1315-1329, 2015.

[4] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Using Deep Learning to Detect Price Change Indications in Financial Markets," *European Signal Processing Conference, Greece*, 2017.

[5] P. Weber and B. Rosenow, "Order book approach to price impact," *Quantitative Finance*, vol. 5, no. 4, pp. 357-364, 2005.

[6] F. Abergel, M. Anane, A. Chakraborti, A. Jedidi, and I. Toke, "Limit Order Books," *Cambridge University Press*, 2016.

[7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751, 2014.

[8] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *arXiv 1510.03820*, 2015.