

深層学習と高頻度データを用いた株式注文状況の推定

田代 大悟^{1*} 和泉 潔¹
Daigo Tashiro¹ Kiyoshi Izumi¹

¹ 東京大学大学院工学系研究科

¹ Graduate School of Engineering, The University of Tokyo

Abstract: For algorithmic trading, it is important to reduce market impact and opportunity costs that closely related to market liquidity. In this work, we propose a tick-based approach to prediction of the liquidity. Our method utilizes order data encoded according to its flexibility and a Long Short-Term Memory(LSTM) that predict a next order. Accuracy of the model outperforms by a large margin maximum occurrence ratio of order labels. Furthermore, we examine the embedding layer of the trained model and find out that it obtains difference and similarity between each order.

1 はじめに

近年の金融市場では、HFT (High Frequency Trading) やアルゴリズム・トレードといった機械的取引が台頭している。いずれもコンピュータを利用して自動的に発注と売買を行う点で共通するが、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、アルゴリズム・トレードは比較的低頻度にかつより高度な戦略を用いて執行する。マーケット・インパクト、機会コストといった取引コストを考慮しつつ、イントラデイでの大口取引を小口に分割し執行することが求められる。

アルゴリズム・トレードの主な執行戦略としてはVWAP (volume-weighted average price) 戦略 [1] が挙げられる。VWAP 戦略とは、取引執行価格を出来高の加重平均をとった価格に近づくように、大口注文を小口に分割し発注する戦略である。さらに、深層学習による価格予測をVWAPと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る。取引コストをいかに縮小するかは、アルゴリズム・トレード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][3]。

さらにミクロな間隔では、執行自身が与える需給の影響により取引が不利になるコストであるマーケット・インパクトや、板の希薄さによって予定した価格での執行ができないリスクである機会コストなどに関する研究が行われている。これらのコストには、市場の需給を表す板のスプレッドや厚さ、回復速度といった流動性が大きく影響する。

一方、金融市場の電子化と高速化による注文量の増加にともない、蓄積される注文情報も膨大化しており、大量の学習データが必要なディープラーニングへの応用の土台は整っている。J. Sirignano は、板情報に対するデータドリブンなアプローチとして、オンラインで学習できる点、大量のデータを扱える点、確率を出力できる点をメリットとして指摘した上で、ディープラーニングは最良の選択肢であると述べている [4]。

2 関連研究と本研究の目的

深層学習を用いて市場予測を行った関連研究を紹介する。

N. Du らはマーク付き点過程の強度関数をRNNを用いた非線形関数とすることで、約定の方向とその時刻の予測を行い、通常のRNNより良い成果を得た [5]。J. Sirignano は、板に対してファットテールに強いニューラルネットワークを提案し、ベストアスクとベストビッドの分布を予測し一定の成果を得た [4]。やはり前述の通り、多次元の情報を持ち非線形で複雑に作用する情報をモデリングする場合にディープラーニングは有効であると言える。

取引主体は何らかの情報を得た上で発注する。市場に集まる注文のパターンにはその背後にある情報が反映されると考えられ、深層学習を用いることでそのような注文パターンを獲得できる可能性がある。

そこで本研究では、板の変化量や価格の変化の予測をティックベースで行うことによって、マーケット・インパクトコストや機会コストに影響を与える将来の板の厚さや板の回復度を予測できるモデルの構築と分析を行う。応用として、あらかじめ流動性の減少を予測し、早期に執行を行うといったことなどが考えられる。

*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

3 LSTMを用いた注文状況予測

ここでは、モデルへの入力の前処理であるオーダーの符号化について説明する。次に、LSTMを用いて直後のオーダーを予測するモデルの構築を行う。

3.1 オーダーの符号化

まず前処理の方法について述べる。オーダーの自由度には指値注文/成行注文/キャンセル注文といった注文のタイプの他、価格、注文量などがある。ここではタイプと価格のみに限定する。まず各指値注文に対して、注文の入った時刻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、指値価格と仲値との差の絶対値を保持する。

次に Figure 1 に示すフローに沿って、各注文に識別符号を与える。まず、注文のタイプとして売成行/買成行/売指値/買指値/売キャンセル/買キャンセルに分ける。成行注文二種に関しては、そのまま識別符号を付与し、指値注文とキャンセル注文に関しては、仲値からの価格差によって分類後、符号を付与する。なお、オーダーと識別符号間の変換は一意であり、符号からオーダー、つまりタイプと価格差へは復元可能となっている。

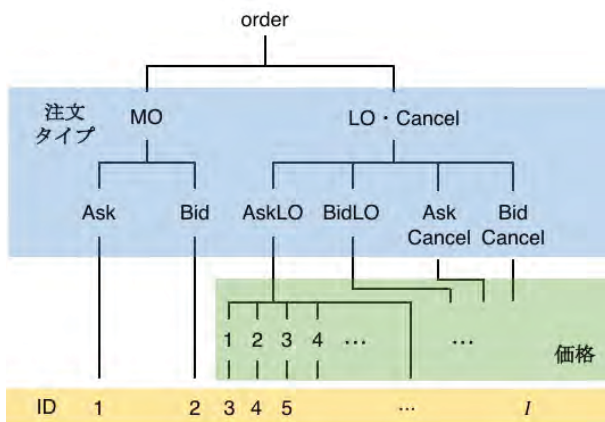


Figure 1: オーダーの符号化フロー。注文タイプ MO/LO/Cancel はそれぞれ/Market Order/Limit Order/Cancel 注文を、Ask/Bid はそれぞれ売り/買いを指す。

3.2 問題設定

本節では、オーダー予測の問題設定を行う。モデルの入力となる系列の集合を $C = \{S^1, S^2, \dots\}$ と表す。各系列 S^i は日毎の寄り付き後から引け前までの前場と後場のいずれかのザラバに対応する。ザラバと板寄せでの注文の入り方では意味が異なるため、ザラバのティックデータだけを扱い、同日の前場と後場は異なる系列

として扱う。ここで、 t 番目のオーダーを x_t とおくと、各系列 S^i は $S^i = (x_1^i, x_2^i, \dots)$ で表される。各オーダー x_t^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、過去のオーダー系列 $(x_1, x_2, \dots, x_{t-1})$ が与えられたときの次のオーダー x_t 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 $p(x_t|x_1, \dots, x_{t-1}; \theta)$ 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。

本研究のモデルは RNN Encoder-Decoder[6][7] を模して設計した。学習後、次のオーダーの予測を繰り返してシミュレートすることで t から T 先までのデコードする場合の条件付き確率分布は以下のようになる。

$$p(x_t, \dots, x_{t+T}|x_1, \dots, x_{t-1}; \theta) = \prod_{t'=t}^T p(x_{t'}|x_1, \dots, x_{t'-1}; \theta) \quad (1)$$

複数オーダーの予測の場合は、ビームサーチを用いればよい。同時確率を最大とするオーダーの系列を探索し、それを最適なオーダー系列とし、さらに板に足し合わせるにより板の変化量や価格を予測することができる。

3.3 モデル

モデルの RNN には、LSTM (Long Short-Term Memory) [8] を用いる。LSTM は、再起的な入出力を行うノードの活性化関数として恒等関数を導入することで、通常の RNN で長い系列データを学習する際に発生する勾配消失問題を回避する。さらに、メモリノードと 3 つの判断ノードから構成される集合体を一つのメモリセルとして扱うことにより表現力の低下を防ぐ。

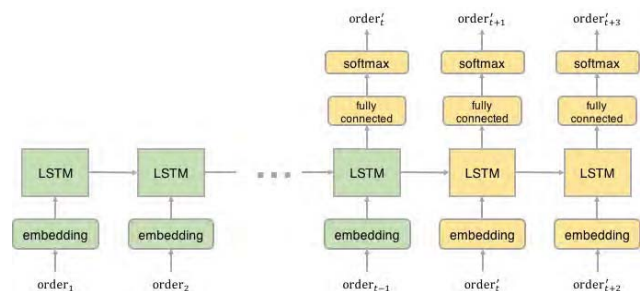


Figure 2: モデルの時系列展開図。

本研究で用いるモデルを時系列で展開した様子を Figure 2 に示す。onehot ベクトルで表される t 時点でのオーダーと embedding 行列 W_{embed} の積をとりオーダーの埋め込みベクトル e_t を得る。 e_t と $t-1$ 時点で

の LSTM セルの出力 h_{t-1} が t 時点での LSTM セルの入力となる。ここで t 時点でのメモリセルのベクトルを c_t , 重みを W_{LSTM} とおくと, LSTM の出力は次のように更新する。

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ g_t \end{bmatrix} = \begin{bmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{bmatrix} W_{LSTM} \begin{bmatrix} h_{t-1} \\ e_t \end{bmatrix} \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (3)$$

$$h_t = o_t \odot \tanh(c_t) \quad (4)$$

さらに, LSTM の出力は全結合層 (W_{full}), softmax 層へと経て出力の条件付き確率を得る。

3.4 学習

(5) 式の負の対数尤度の最小にするようパラメータ $\theta = (W_{embed}, W_{LSTM}, W_{full})$ を探索し, 最適パラメータ $\theta = \theta^{opt}$ を得る。

$$\mathcal{L} = - \sum_{i=1}^C \sum_{t=1}^{S^i} \log p(x_t^i | x_1^i, \dots, x_{t-1}^i; \theta) \quad (5)$$

ただし, $C = |C|$ かつ $S^i = |S^i|$ である。

4 実験

4.1 データセット

データセットには 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。使用する銘柄と期間は大和ハウス工業 (銘柄コード 1925) の 2013 年 7 月 1 日から 2014 年 6 月 30 日までと, キリンホールディングス (銘柄コード 2503) の 2013 年 7 月 1 日から 2014 年 3 月 31 日までとした。各銘柄のデータの日数を 8:2 に分割し, 日時の古い期間のデータを学習データ, 新しい期間のデータをテストデータとする。

4.2 前処理

3.1 で述べた手法で, オーダーの前処理を行う。本実験では, オーダーの自由度を注文タイプと価格のみに限定する。価格の深さの自由度は 20 として, それ以上の価格に入ったオーダーは 20 として分類した。前処理後, オーダーの自由度を表す識別符号のサイズ I は $I = 82$ となる。分類後の主なオーダーの出現回数比を Table 1 に示す。

Table 1: 銘柄 1925 における分類比率 (比率による降順)。

分類	注文タイプ	価格差	比率
	BidLO	1	0.076
	AskLO	1	0.074
	BidCancel	1	0.058
	AskCancel	1	0.057
82 分類 (注文タイプ+価格)	BidLO	2	0.056
	AskLO	2	0.056
	⋮		
	AskMO	—	0.019
	BidLO	5	0.018
	BidMO	—	0.018
	⋮		
	BidLO	—	0.259
	AskLO	—	0.251
6 分類 (注文タイプ)	BidCancel	—	0.231
	AskCancel	—	0.223
	AskMO	—	0.019
	BidMO	—	0.017

4.3 実装

(5) 式が最小となるように, バッチサイズを 5 としたミニバッチを用いた確率的勾配降下法で最適化を行う。同系列内では LSTM のメモリセルの状態は保持しつつ, 新しい系列セットを用いる際に状態を初期化する。学習時には, 100 オーダー毎に勾配情報を破棄した。ハイパーパラメータは epoch 数 50, 隠れ層のユニット数は全て 40, ドロップアウトの比率 0.5, LSTM の層 1 とした。深層学習フレームワークとして chainer を使用し実装した。

5 結果と考察

5.1 モデルの評価

テストデータを用いて, 学習済みモデルによる予測と評価を行う。まず, LSTM のメモリセルの状態を得るため, 予測を行わず先頭から 100 オーダーを入力する。次に, 残りの系列を用いて 1 オーダーごとの予測と評価を逐次的に行う。モデルは直後の 1 オーダーのみを未知のものとし, Encoder-Decoder モデルのごとく出力を自らの入力として与えることはここでは行わない。

結果の評価は, 前処理での分類による 82 分類そのままの評価と, 6 分類での評価を行う。82 分類では, 出

力で得られる条件付き確率の最大となる符号を採用し、正解ラベルと比較する。6分類評価では、その82分類で予測した注文のタイプと価格のうち注文のタイプだけを対象とした6分類での評価を行う。例えば、得られた予測がタイプ AskLO で価格差2で正解ラベルがタイプ AskLO で価格差3の場合、82分類では誤と判別するのにに対し、6分類評価では正と評価する。6分類評価では、粒度を粗くすることによって、モデルが注文の方向性に対してどの程度説明できるかを評価する。

結果を Table 2 に示す。82分類と6分類いずれも、Table 1 に示す出現回数最大の比率を上回る。このことから、モデルは十分に学習しており、オーダーの予測は可能であると言える。

Table 2: 評価方法と精度。

評価方法	銘柄	精度
82 分類	1925	0.201
	2503	0.240
6 分類	1925	0.460
	2503	0.439

5.2 隠れ層のクラスタリング

次に embedding 層の重みを分析する。embedding 層の各オーダーに対応する重みは、embedding 層での各オーダーを表す固有のベクトルに対応する。この40次元82個のベクトルに主成分分析を施す。得られた第一主成分から第十主成分までを用いて k-means クラスタリングを行う。結果を第一主成分と第二主成分上にプロットしたものを Figure 3 に示す。

まず注文のタイプの全体的な分布だけを見ると、 $x = -0.3$ 付近を境界として売り側/買い側で大まか分類できていると言える。特にキャンセルは、指値/キャンセルと売り/買いでいずれもはっきりと分かれる。さらに PC1, PC2 上では線形の関係にあるように見える。

クラスタリングに関しては、キャンセルは売り/買いだけでなく価格の大小によっても分かれている。一方、指値注文は売り/買いに分類できていない。ここから、モデルが指値に関する情報よりもキャンセルに関する情報を捉えていると解釈できる。ただ、このクラスタリングは embedding 層の重みの PC1 と PC2 のみによるものであり、複雑で非線形なモデルであるニューラルネットワークでは、さらに上の層で指値の情報を識別する機構がある可能性もあり、分析の余地が残る。

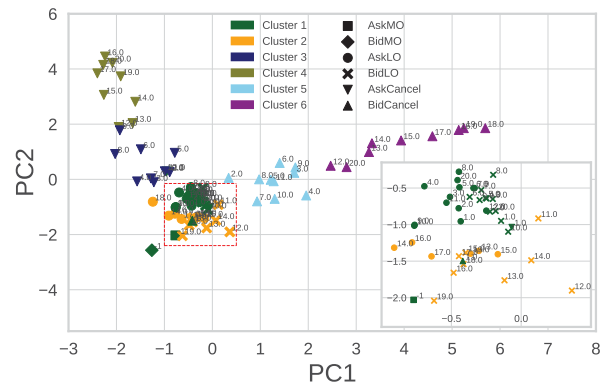


Figure 3: embedding 層の主成分分析とクラスタリングの結果。PC1: 第一主成分, PC2: 第二主成分。各プロットに対応する数字は各指値/キャンセル注文の仲値からの価格差である。成行注文は-1として表現する。プロットの形は注文のタイプを表し、色はクラスタリングの結果を表す。図中右下の図は、赤破線枠の指値の注文が多く集まる範囲の拡大図。

6 まとめ

本研究では、オーダーの予測を行い、さらにニューラルネットワークの重みからオーダーの類似性を見ることができた。具体的には、前処理としてオーダーの符号化を行い、LSTM を用いて次のオーダーを予測するモデルの学習・予測の実験を行った。さらに学習済み embedding 層の重みを分析することで、各重みに対応するオーダーのベクトルのクラスタリングを行った。今回のオーダーの予測以外の問題でも、オーダーベースでニューラルネットワークを学習することによって、オーダーの関係性や類似度が得られる可能性がある。問題設定によって、その関係性がどのように変わるかは興味深いテーマだと思われる。

本研究のモデルへの入力、板の状態に関する情報を含めない、オーダーベースでかつ価格情報は仲値からの価格差を用いた。しかし板の状態によってオーダーの意味は変わると考えられ、改善の余地がある。例えば、同じ価格差のオーダーであっても、スプレッドが大きい時ではスプレッドの中に入り得、逆ではスプレッドの外に入ることになる。また 100 円に対する 1 円と 1000 円に対する 1 円とでは意味が異なるように、価格の大きさに対する価格差が持つ意味も異なる。

これを踏まえて今後の展望としては、板の状態やオーダーの量、時間差といった入力情報の拡充、価格差の比率によって前処理の分類を行うなどの工夫が挙げられる。

参考文献

- [1] R. Almgren and J. Lorenz. Adaptive Arrival Price. *Algorithmic Trading III, Institutional In-*

vestor, 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] 杉原慶彦 「執行戦略と取引コストに関する研究の進展」『金融研究』第31巻第1号, 227–292頁, 2012年.
- [4] J. Sirignano . Deep learning for limit order books, *arXiv 1601.01987v7*, 2016.
- [5] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent Marked Temporal Point Processes Embedding Event History to Vector. *Knowledge Discovery and Data Mining*, 2015.
- [6] I. Sutskever, O. Vinyals, and O. Le. Sequence to sequence learning with neural networks. *Neural Information Processing Systems*, pp. 3104–3112. 2014.
- [7] K. Cho, B. Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the Empirical Methods in Natural Language Processing*, 2014
- [8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, pp. 1735–1780. 1997.