
Extending Deep Learning Models for Limit Order Books to Quantile Regression

Zihao Zhang¹ Stefan Zohren¹ Stephen Roberts¹

Abstract

We showcase how Quantile Regression (QR) can be applied to forecast financial returns using Limit Order Books (LOBs), the canonical data source of high-frequency financial time-series. We develop a deep learning architecture that simultaneously models the return quantiles for both buy and sell positions. We test our model over millions of LOB updates across multiple different instruments on the London Stock Exchange. Our results suggest that the proposed network not only delivers excellent performance but also provides improved prediction robustness by combining quantile estimates.

1. Introduction

Traditional time-series modelling is often dominated by Markov-like models with stochastic driving terms such as the vector autoregressive model (VAR) (Zivot & Wang, 2006). These models make strong parametric assumptions to the functional form of the predictive model (in particular the AR family) and also require the target time-series to be stationary. However, financial price (and volume) rarely conform to these assumptions and even returns, the first order differences of prices, are rarely stationary (Cont & Nittens, 1999). Deep learning has gained popularity in financial modelling since they are not constrained by the above assumptions (see (Tsantekidis et al., 2017a;b) for some examples). Modern deep learning architectures also allow one to tailor the loss function, as is demonstrated in (Lim et al., 2019) where the Sharpe ratio is directly maximised as a cost function.

Deep networks often require a large number of observations to calibrate weights and this property fits nicely with finan-

cial applications that utilise high frequency microstructure data. Nowadays, billions of market data are generated everyday and most of them are recorded in Limit Order Books (LOBs) (Parlour & Seppi, 2008; Bouchaud et al., 2018). A LOB is a record of all unmatched orders of a given instrument in a market comprising of levels at different prices containing resting limit orders to sell and buy, also called ask and bid orders. A bid (ask) order is an order to buy (sell) an asset at or below (above) a specified price. We can consider LOBs as the most granular financial data as a LOB represents the demand and supply of a given instrument at any moment in time.

In our previous works (Zhang et al., 2018; 2019), we demonstrate that deep learning models can deliver improved predictive performance, in comparison with standard methods when modelling LOB data. One of the important contributions of deep learning is the ability to automate the process of feature extraction. In the work of (Sirignano & Cont, 2018) as well as our own, it is demonstrated that such models can extract representative features that are related to the demand and supply in the order book. Given billions of market quotes from LOBs, this approach has proved to be more effective than algorithms that rely on hand-crafted features. Features derived from a human-centric understanding of a process (such as “moving-average crossover”) do not guarantee best performance with respect to the target function, as is also seen in (Lim et al., 2019). Further, in complex non-stationary environments such as finance, it is far from trivial to select informative features by hand, even after years of work in the industry.

In this work, we utilise deep neural networks and Quantile Regression (QR) (Koenker & Hallock, 2001) to model the returns from LOBs. QR is particularly useful for financial time-series as it models the conditional quantiles of a response. Returns are in general heterogeneous, highly peaked and have fat tails compared to a normal distribution (Cont & Nittens, 1999). A point estimation is really not enough to describe the full distribution of returns and we can obtain considerably more information by estimating multiple quantiles. We regard Quantile Regression as able to provide valuable, non-stationary, extra information regarding risk exposure.

¹Oxford-Man Institute of Quantitative Finance, Department of Engineering Science, University of Oxford, Oxford, United Kingdom. Correspondence to: Zihao Zhang <zihao@robots.ox.ac.uk>.

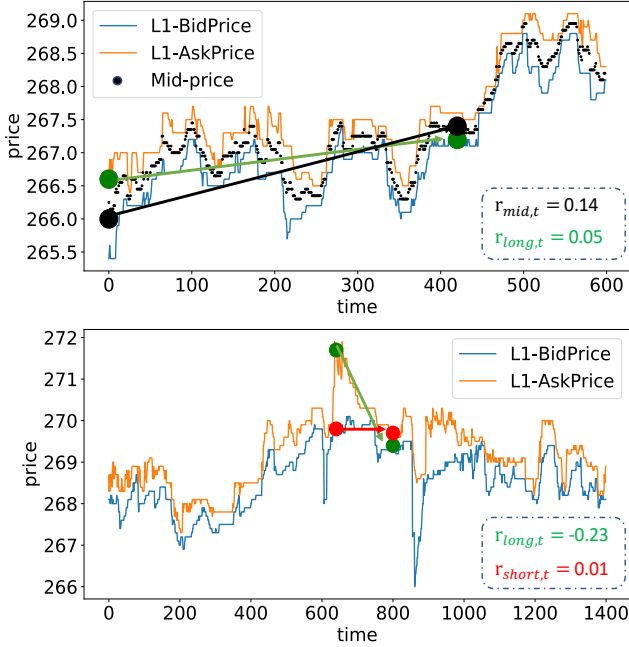


Figure 1. **Top:** returns r obtained from using mid-price or first level prices from LOBs. **Bottom:** returns obtained from long and short positions at a given time stamp.

Unlike most literature, where mid-price is used to represent financial time-series, we define returns by directly using first level prices from LOBs. Modelling mid-price is appropriate if we are using daily data, but it is improper for intraday strategies as we ignore spreads which are the differences between best ask and bid price. The upper plot of Figure 1 illustrates the effects of spreads as transaction costs on returns. The return, $r_{mid,t}$, from using mid-price is about three times higher than the actual return, $r_{long,t}$, that we can obtain (we have to buy from ask sides and sell from bid sides for aggressively entering or exiting positions). Further, intraday strategies often involve both long and short positions to increase profitability. However, returns from these two positions are not symmetric at a given time stamp. We observe this at the bottom of Figure 1. A return, $r_{long,t}$, is -0.23 from a long position, but it does not imply a profit of 0.23 by taking a short position ($r_{short,t} = 0.01$). Indeed, the two return series are statistically different under Kolmogorov-Smirnov and Wilcoxon signed-rank tests. This discrepancy comes from changing spreads and indicates that separate models are required to estimate returns for different positions.

Our contributions: We propose a network architecture that can simultaneously estimate multiple return quantiles from both buy and sell positions by training with different Quantile Loss functions. Our model consists of a block of con-

volutional layers and multiple LSTM branches to estimate different quantiles. The convolutional block, as a feature extraction mechanism, processes raw limit order book data and LSTM layers are used to capture time dependencies among the resulting feature maps. We show that this method delivers better predictive performance than other popular machine learning algorithms. Also, better performance can be achieved by combining estimates from different quantiles.

2. Data Description and Returns

Our dataset consists of one year full-resolution LOB data for five of the most liquid stocks listed on the London Stock Exchange (LSE), namely, Lloyds Bank, Barclays, Tesco, BT and Vodafone. The data spans all trading days from the 3rd of January 2017 to the 24th of December 2017 and only normal trading periods (between 08:30:00 and 16:30:00) are included. We take price and volume for 10 levels on both ask and bid sides of a LOB so there are 40 features at each timestamp. Overall, our dataset has more than 134 million observations and there are, on average, 150,000 events per day per stock. The first 6 months are used as training data, the next 3 months as validation data and the last 3 months as test data. In the context of high-frequency data, 3 months test data corresponds to millions of observations and therefore provides sufficient scope for testing model performance and robustness.

We prepare our input data using the procedure outlined in our previous work (Zhang et al., 2019). We define returns by taking spreads into account. A return, $r(t)$, at time t , both long or short, can be decomposed as:

$$r_i(t) = \frac{\Delta p_i(t)}{p_{mid}(t)}, \Delta p_i(t) = z_i(t) \Delta m(t) - \frac{s(t) + s(t+k)}{2} \quad (1)$$

where

$$\begin{aligned} z_i(t) &= \begin{cases} 1, & \text{if } i = \text{long} \\ -1, & \text{if } i = \text{short} \end{cases} \\ \Delta m(t) &= p_{mid}(t+k) - p_{mid}(t) \\ p_{mid}(t) &= \frac{p_{ask}^{(1)}(t) + p_{bid}^{(1)}(t)}{2} \end{aligned} \quad (2)$$

and k is the predicted horizon, $s(t)$ and $s(t+k)$ are spreads at time t and $t+k$. We denote the first level ask and bid price as $p_{ask}^{(1)}(t)$ and $p_{bid}^{(1)}(t)$. A schematic description of Equation (1) is given in Figure 2 – we cross half the spread now, follow the mid price and cross the other half later. As we already observe the current spread $s(t)$, there is no need to model it and we take it out from (1), so the return $r'_i(t)$ of interest is defined as:

$$r'_i(t) = \frac{\Delta p'_i(t)}{p_{mid}(t)}, \quad \Delta p'_i(t) = \Delta p_i(t) + s(t). \quad (3)$$

Note that Equation (3) can be also written as:

$$r'_i(t) = z_i(t)r_{mid}(t) - r_{spread}(t)/2 \quad (4)$$

where $r_{mid} = \Delta m(t)/p_{mid}(t)$ and $r_{spread}(t) = (s(t+k) - s(t))/p_{mid}(t)$. Instead of modelling $(r'_{long}(t), r'_{short}(t))$, we can also estimate quantiles for mid-price change and spread change $(r_{mid}(t), r_{spread}(t))$.

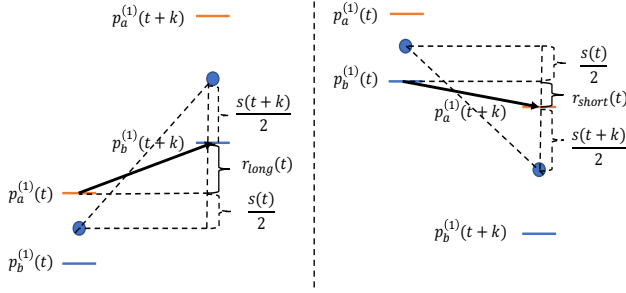


Figure 2. A schematic description of Equation (1). $p_a^{(1)}(t)$ and $p_b^{(1)}(t)$ represent best ask and bid prices at time t . **Left:** Shows a return from a long position; **Right:** Shows a return from a short position.

3. Methods and Network Architecture

3.1. Quantile Regression

In Quantile Regression (QR), we predict the conditional quantile of the target distribution, i.e. $\mathbb{P}(r \leq r^{(\tau)} | x) = \tau$, where τ is the quantile of interest and x is the input. We can obtain the estimates by minimizing the Quantile Loss (QL) with respect to $r^{(\tau)}$, leading to the τ th quantile:

$$L_\tau(r, \hat{r}^\tau) = \sum_{t:r(t) < \hat{r}^\tau(t)} (\tau - 1)|r(t) - \hat{r}^\tau(t)| + \sum_{t:r(t) > \hat{r}^\tau(t)} \tau|r(t) - \hat{r}^\tau(t)| \quad (5)$$

where $r(t)$ is the observation at time t and $\hat{r}^\tau(t)$ is the predicted value. The common used mean absolute error is equivalent to QL with $\tau = 0.5$. Note that each quantile has its own QL function and, in order to obtain multiple quantiles, we need separate models to estimate each of them. Our network solves this constraint by training with multiple QLs to model all quantiles of interest simultaneously.

QR has a common problem known as quantile crossing as quantile curves can cross each other, leading to an invalid distribution for the response, e.g. the predicted 90th percentile of the response is smaller than the 80th percentile which is impossible. We follow the work of (Chernozhuikov et al., 2010) to rearrange the original estimated non-monotone curve into a monotone rearranged curve.

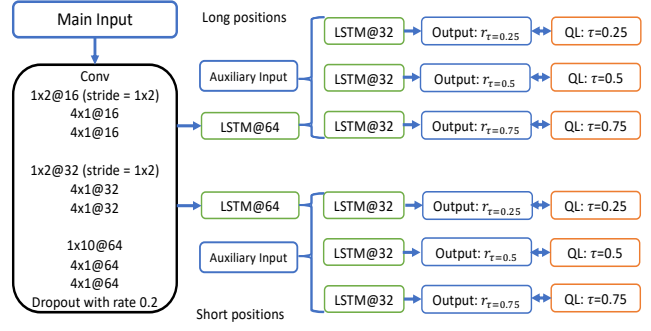


Figure 3. Model architecture for DeepLOB-QR. Here $1 \times 2@16$ represents a convolutional layer with 16 filters of size (1×2) . QL: τ represents the quantile loss at quantile τ .

3.2. Network Architecture

We propose a neural network architecture that simultaneously models different quantiles of returns from both long and short positions. This model is an extension of our previous work (Zhang et al., 2019) and, for a detailed discussion on how initial layers are constructed, please find more information there. We denote this new architecture as DeepLOB-QR.

As each quantile has its own loss function, if we are interested in 3 quantiles for returns from each position, we would need to estimate 6 separate models. This is computationally demanding for LOBs data as we have millions of them in a single day. Further, each of the models is essentially estimating the “same” underlying quantity (long or short return), just different quantiles of it. We would thus expect that there are common features that can contribute to the estimation of all models and it is wasteful to estimate them separately.

DeepLOB-QR is designed to solve the above constraints by using a common convolutional block and branching out several LSTM layers to model different interested quantiles. The “main input” takes raw LOBs data to extract features that can modulate relationships between demand and supply of an instrument. The two auxiliary inputs take past returns from long and short positions. We isolate LOBs data and past returns here because one important property of convolutional layer is parameter sharing and price and volume series have different dynamics compared to returns. Overall, this is a multi-input and multi-output setup but trained using different loss functions (6 QLs in our case). The last parallel LSTM layers (LSTM@32) are only trained using their corresponding losses, while each of the two LSTM@64 layers is trained using 3 losses and the convolutional block is trained using all losses.

3.3. Forecast Combination

The works of (Rapach et al., 2010; Meligkotsidou et al., 2014) suggest that combinations of individual quantile estimates can form a much robust point estimation and help reduce prediction uncertainty. After obtaining quantile estimates $\hat{r}^{(\tau)}(t)$, $\tau \in \mathcal{S}$ where \mathcal{S} denotes the set of considered quantiles, we can combine them as:

$$\hat{r}(t) = \sum_{\tau \in \mathcal{S}} \pi^{(\tau)} \hat{r}^{(\tau)}(t), \quad \sum_{\tau \in \mathcal{S}} \pi^{(\tau)} = 1 \quad (6)$$

where the weights $\pi^{(\tau)}$ represents the probability assigned to the prediction of quantile τ . This estimator, as a linear combination of order statistics, forms a point estimation of the central location of a distribution based on small sets of quantile estimates. We can reflect our beliefs on how each quantile estimate affects the central location by adjusting the corresponding weights.

The simplest method of estimating the weights is to use a fixed weighting scheme. We can also form a constrained optimization problem (Meligkotsidou et al., 2014) to find an optimal combination of quantile estimates:

$$\pi = \arg \min_{\pi} E_t[r(t-1) - \sum_{\tau \in \mathcal{S}} \pi^{(\tau)} \hat{r}^{(\tau)}(t-1)]^2, \quad (7)$$

where $\pi = [\pi^{(\tau)}]_{\tau \in \mathcal{S}}$ and $\sum_{\tau \in \mathcal{S}} \pi^{(\tau)} = 1$. This procedure hence treats quantile estimation as constrained regression, where the weights are non-negative and sum to unity.

4. Experiments and Results

As in Figure 3, we predict three quantiles (0.25, 0.5 and 0.75) for returns from long positions and same three quantiles for short positions, totalling 6 QLs. Our prediction horizon, k , is 100 steps into the future. In order to compare with other methods, we assess how different models estimate the central location of a response. We denote estimates of the 0.5 quantile from our model as DeepLOB-QR and estimates obtained from the combination scheme (Equation (7)) as DeepLOB-QR(C). We compare to four other models: an autoregressive model (AR), a generalised linear model (GLR), a support a vector regression (SVR) and a neural network with multiple fully connected layers (MLP).

Table 1 shows the results of our model compared to other methods. Performance is measured by mean absolute error (MAE), mean squared error (MSE), median absolute error (MEAE) and R^2 score. All errors, except R^2 score, are divided by errors from a repetitive model¹, therefore, we understand how much better a model is compared to a zero-intelligence method. DeepLOB-QR achieves the smallest errors and the highest R^2 for modelling returns from

Table 1. Experiment results for the LSE dataset.

MODEL	MAE	MSE	MEAE	R^2
METRICS FOR RETURNS FROM LONG POSITIONS				
AR	0.705	0.466	0.793	0.004
GLR	0.719	0.556	0.707	-0.188
SVR	0.802	0.498	0.755	-0.063
MLP	0.717	0.466	0.708	0.008
DEEPLQOB-QR	0.701	0.462	0.710	0.010
DEEPLQOB-QR(C)	0.701	0.461	0.702	0.014
METRICS FOR RETURNS FROM SHORT POSITIONS				
AR	0.708	0.498	0.791	0.004
GLR	0.736	0.528	0.709	-0.055
SVR	0.738	0.548	0.708	-0.096
MLP	0.726	0.510	0.708	0.002
DEEPLQOB-QR	0.705	0.501	0.720	0.013
DEEPLQOB-QR(C)	0.702	0.494	0.699	0.015

Table 2. Experiment results for different prediction horizons k .

MODEL	MAE	MSE	MEAE	R^2
METRICS FOR RETURNS FROM LONG POSITIONS				
DEEPLQOB-QR _{k=50}	0.626	0.524	0.521	0.025
DEEPLQOB-QR(C) _{k=50}	0.634	0.514	0.513	0.042
DEEPLQOB-QR _{k=100}	0.701	0.462	0.710	0.010
DEEPLQOB-QR(C) _{k=100}	0.701	0.461	0.702	0.014
DEEPLQOB-QR _{k=200}	0.701	0.417	0.864	0.010
DEEPLQOB-QR(C) _{k=200}	0.719	0.419	0.841	0.011
METRICS FOR RETURNS FROM SHORT POSITIONS				
DEEPLQOB-QR _{k=50}	0.624	0.493	0.494	0.024
DEEPLQOB-QR(C) _{k=50}	0.644	0.483	0.501	0.044
DEEPLQOB-QR _{k=100}	0.705	0.501	0.720	0.013
DEEPLQOB-QR(C) _{k=100}	0.702	0.494	0.699	0.015
DEEPLQOB-QR _{k=200}	0.700	0.428	0.864	0.012
DEEPLQOB-QR(C) _{k=200}	0.718	0.430	0.837	0.011

both long and short positions. Also, our results suggest that better prediction results can be obtained from forecast combinations (DeepLOB-QR(C)). However, as the prediction horizon (k) increases, the problem becomes more difficult and we obtain worse performance, as shown in Table 2. To visualise how quantile estimates form a prediction interval, we plot them against the true observations in Figure 4.

5. Conclusion

In the context of time-series from high-frequency limit order book (LOB) data we show that quantile regression (QR) can provide us with prediction uncertainty by forming a confidence interval and better predictive performance can be obtained by combining multiple quantile estimates. A promising direction for future work is to design trading

¹A repetitive model means using current observations as future predictions, a zero-intelligence model.

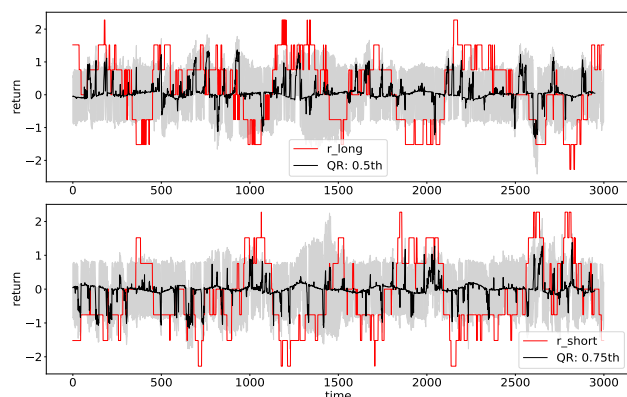


Figure 4. Top: Real returns from long positions is in red and 0.5 quantile estimate is in black. The upper boundary of grey shading represents 0.75 quantile estimate and the lower boundary is for 0.25 quantile estimate; **Bottom:** Real returns and quantile estimates for short positions.

strategies based on derived signals. We can combine QR with Reinforcement Learning and use uncertainty information to tackle the exploration and exploitation problem.

References

- Bouchaud, J.-P., Bonart, J., Donier, J., and Gould, M. *Trades, quotes and prices: Financial markets under the microscope*. Cambridge University Press, 2018.
- Chernozhukov, V., Fernández-Val, I., and Galichon, A. Quantile and probability curves without crossing. *Econometrica*, 78(3):1093–1125, 2010.
- Cont, R. and Nitiens, D. Statistical properties of financial time series. 1999.
- Koenker, R. and Hallock, K. F. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, 2001.
- Lim, B., Zohren, S., and Roberts, S. Enhancing time series momentum strategies using deep neural networks. *arXiv preprint arXiv:1904.04912*, 2019.
- Meligkotsidou, L., Panopoulou, E., Vrontos, I. D., and Vrontos, S. D. A quantile regression approach to equity premium prediction. *Journal of Forecasting*, 33(7):558–576, 2014.
- Parlour, C. A. and Seppi, D. J. Limit order markets: A survey. *Handbook of financial intermediation and banking*, 5:63–95, 2008.
- Rapach, D. E., Strauss, J. K., and Zhou, G. Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies*, 23(2):821–862, 2010.
- Sirignano, J. and Cont, R. Universal features of price formation in financial markets: Perspectives from deep learning. Available at SSRN 3141294, 2018.
- Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., and Iosifidis, A. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 1, pp. 7–12. IEEE, 2017a.
- Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., and Iosifidis, A. Using deep learning to detect price change indications in financial markets. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2511–2515. IEEE, 2017b.
- Zhang, Z., Zohren, S., and Roberts, S. BDLOB: Bayesian Deep Convolutional Neural Networks for Limit Order Books. In *Third workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018.
- Zhang, Z., Zohren, S., and Roberts, S. DeepLOB: Deep Convolutional Neural Networks for Limit Order Books. *IEEE Transactions on Signal Processing*, 2019.
- Zivot, E. and Wang, J. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.