Predicting Intraday Price Movements in the Foreign Exchange Market

Noam Brown

Robert Mundkowsky

Sam Shiu

Abstract

It is commonly assumed that short-term price movements follow a random walk and cannot be predicted. However, in this project we predict next-second price movements in the euro-dollar foreign exchange market by using depth as a feature. We show that if there is a sufficient imbalance in depth, an accurate prediction can be made. Further, we train and test a Markov Model to demonstrate that this predictive potential can overcome transaction costs to be profitable.

1. Data

Our data consists of a 1-second time series of prices and quotes in the euro-dollar foreign exchange market. All prices are measured in dollars per euro. The data spans from 1 February 2008 to 13 March 2008, resulting in about 2.5 million observations. Since the foreign exchange market is open 24 hours a day 5 days a week, we have access to many millions of observations. The earlier 70% of the observations were used for training, with the remaining 30% used for predictions.

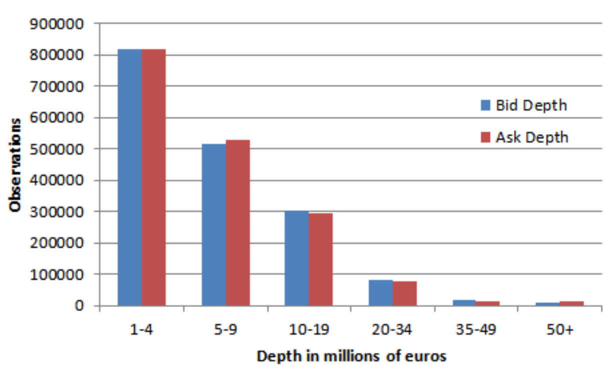

Figure 1. Distribution of magnitude of depth.

Unlike most financial market data, our dataset includes a complete aggregate of the bid and ask depth at each second, measured in millions of euros. That is, we know the combined size of standing requests to buy (bid) or sell (ask) at almost any given price. This data will form the basis for the key features we use in our predictions.

A price decrease can only occur after all the bid depth at a given level is exhausted. Likewise, a price increase can only occur after all the ask depth at a given level is filled. It seems logical, then, that market depth would influence the short-term movement of prices. Suppose the current best bid price at time t (measured in seconds) is p_{b_t} with depth d_{b_t} and the current best ask price at time t is p_{a_t} with depth d_{a_t}. Intuitively, one would think that if $d_{b_t} >> d_{a_t}$, then $P(p_{b_{t+1}} < p_{b_t}) < P(p_{a_{t+1}} > p_{a_t})$. That is, if there are far more people trying to buy at the bid than there are people trying to sell at the ask, then we expect the bid is relatively less likely to go down in price than the ask is likely to up in price. This is because there is less resistance to an upward movement than a downward movement. This will be the central approach to our predictions.

1.1. Feature selection

Following the convention of the literature, instead of attempting to predict the next transaction price we instead attempted to predict the next mid-quote. This mid-quote gives a more stable measure of price move-

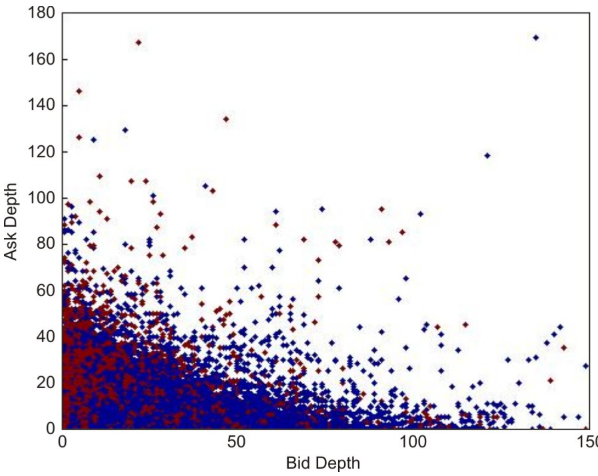

Figure 2. Random sampling of upticks (blue) vs downticks (red) across depth. Note that downticks are more prevalent when ask depth is higher.

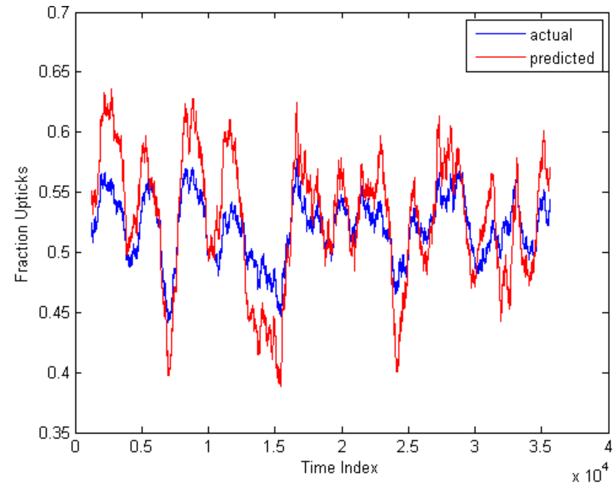

Figure 3. A moving average of perceptron predictions of next-second price movements using only bid and ask depth as features.

ments in the market.[1] It is calculated by averaging the highest bid and the lowest ask. For convenience, we will still use p to represent the mid-quote and refer to it as the price. Since we know what the current price is, we actually need only predict the change in mid-quote Δp. This change in price is measured in pips, which are ten-thousandths of a point[2]. 50.1% of these changes are positive in the training data, while 49.9% are negative.

We included a total of three features in our predictions. Two critical features are depth at the highest bid d_b, and depth at the lowest ask d_a. Previous literature has shown that at high frequencies, there is some mean reversion. We confirmed this by running an autoregression on the mid-quote series. Therefore, we also included the previous change in price Δp_{t-1} as a feature in the Markov model.

We also considered using other available features, including the spread[3], trade volume in the last second, trade volume in the last minute, and Δp_{t-2}. However, none of these features contained significant predicting power.

[1]Trades occur anywhere between the highest bid and the lowest ask, inclusively. Thus, even if the bid and the ask do not move, oscillations in trade prices between the bid and the ask can give the illusion of movement.

[2]An increase in the exchange rate from 1.3145 to 1.3146 is an increase in one pip.

[3]The distance between the highest bid and the lowest ask.

2. Perceptron model

As a first step, we attempted to simply predict the direction of future price movements using a perceptron model. We scored our results only on seconds when there was an actual price movement. Without any further constraints, this method obtained a prediction accuracy of 69%.

However, in the market it is not necessary to trade constantly. Instead, one can choose to only trade when there is a certain degree of confidence. With this in mind, we restricted our predictions to only scenarios where there is a significant depth imbalance. We chose a threshold of 10 (million euros). With this further restriction, our model achieved a prediction accuracy of 81% making predictions on 25% of observations that had a price movement.

We also attempted predictions on time horizons greater than 1 second. However, the accuracy of our predictions fell off quickly as we moved beyond 1 second, as shown in figure 4. At 2 seconds, the accuracy fell to 64%, at 3 second it fell to 57%, and by 7 seconds our features had no predicting power. This is reflective of the fast pace of the foreign exchange market. Indeed, 1 second itself is considered a large interval by traders.

3. Markov model

For our perceptron algorithm, we restricted our predictions to only situations where we were confident in our results. This raises the obvious question - what confidence threshold is optimal? To answer this ques-

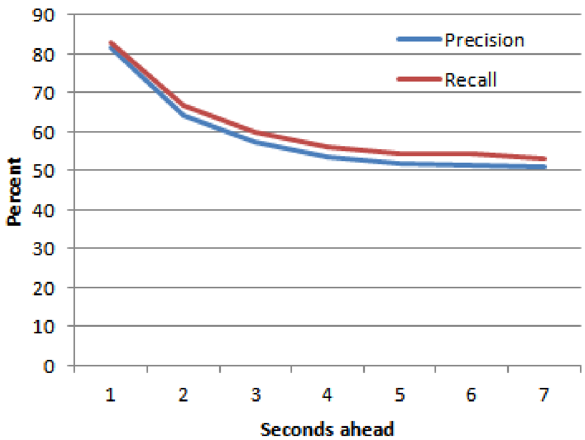

Figure 4. Accuracy attempting to predict over different time horizons. Performance falls off quickly.

tion, we constructed a Markov model with rewards and costs to simulate market conditions.

Since we incorporated costs for acting on a prediction, it no longer made sense to evaluate a model based on prediction accuracy alone. After all, the model may accurately predict a price movement, but believe that the transaction cost of acting on that information outweights the expected value gain of the movement itself. Instead, we evaluated our model based on a theoretical profit, considering both rewards and costs, that the model would seek to maximize.

3.1. State space

Our Markov model contains a 4-dimensional state space consisting of bid depth, ask depth, change in price since last second, and position. Position can take one of three discrete values: long, neutral, and short. This represents whether we have bought euros, sold euros, or do not hold any euros. The position dimension is obviously central to our reward calculations. Bid depth and ask depth were discretized into 6 buckets each. Change in price since last second was discretized into 3 buckets: positive, negative, and zero. This amounts to 324 different possible states.

Due to the very high liquidity in the euro-dollar foreign exchange market and the small quantities our model trades, we can safely assume that our actions do not significantly affect the market. That is, whether we are long, short or neutral has no bearing on if the price goes up or down, or if the depth increases or decreases. Therefore, the position state dimension is independent of the other three. Since the dimension is independent, it need not be included in the transition probability matrix. This reduced our transition

	Long	Neutral	Short
Buy	N/A	Long	Neutral
Nothing	Long	Neutral	Short
Sell	Neutral	Short	N/A

Table 1. Action/position transition matrix.

	Long	Neutral	Short
Buy	N/A	-0.5	-0.5
Nothing	0	0	0
Sell	-0.5	-0.5	N/A

Table 2. Action cost matrix.

	Long	Neutral	Short
Uptick	0.70	0	-0.70
Nothing	0	0	0
Downtick	-0.70	0	0.70

Table 3. Position reward matrix.

matrix to 108x108. Our transition probability matrix was created by observing the 1.7 million transitions among these states in the training data. Every state was reached at least once.

As a small test, we repeated our attempt to predict the direction of price movements, this time using the Markov probability matrix. That is, given a bid depth b_t, ask depth a_t, and previous price movement l_t, we calculated the corresponding state s_t, and checked whether we were more likely to transition upward (to a state with a positive previous price movement $l_{t+1} > 0$) or downward (to a state with a negative previous price movement $l_{t+1} < 0$). That is, we compared $P(l_{t+1} > 0)$ to $P(l_{t+1} < 0)$ and predicted whichever was more likely. To improve our confidence in the result, we required a depth imbalance such that $b_t \neq a_t$ in order to make a prediction. Testing on our prediction set, we achieved 77% accuracy making predictions on 68% of the observations that had price movements, and 13% of all observations. Consider the number of predictions made, this is about equivalent to our results with the perceptron algorithm. Although some information was lost in the bucketing of the depth, it is possible that the ability to model the data as non-linear countered that loss.

We also included 3 actions: buy, sell, or do nothing. Buy and sell will transition the position as shown in Table 1. We assume that each purchase is for 1 million euros, the smallest amount allowed in this market. To keep our market impact assumption realistic, we disallowed the model from further buying in a long position or further selling when in a short position.

3.2. Rewards and costs

Even though we are able to predict the direction of price change with greater than 50% accuracy, it may often not be profitable to act on this information. This is because there are inherent transaction costs in any purchase. The most significant transaction cost, particularly when trading at high frequencies, is the spread.[4] In the euro-dollar market, the spread at normal trading hours is almost always 1 pip. Therefore, we assumed that any trade has an immediate cost of 0.5 pips, so that a buy and a sell combined equals the cost of a typical spread. Since our transactions are always to purchase 1 million euros, this cost amounts to \$50. This is shown in Table 2.

Our reward function was chosen to resemble profitability in the market. If we are in a long position, then we profit from upticks (positive changes in price) and lose money on downticks. Similarly, if we are in a short position then we profit from downticks and lose money on upticks. The models knows if it is long or short based on the position dimension of the state. Likewise, the model knows if there has just been an uptick or downtick based on the "change in price since last second" dimension of the state. Thus, the reward for any state can be determined from these dimensions, as shown in the Table 3. To simplify the training of our model, all rewards have a magnitude of 0.70 pips (\$70). This is the average magnitude of upticks and downticks in our data set.

Thus, the value function for state s_t defined by its dimensions: position p_t, bid size b_t, ask size a_t, and last price move l_t is given by

$$V_{s_t} = R_{p_t,l_t} + \gamma \sum_s [P(b,a,l|b_t,a_t,l_t)(V_s - C_{p,p_t})]$$

Where p, b, a, and l are position, bid depth, ask depth, and last price respectively; R_{p_t,l_t} is the reward for being in position p_t when there was a price movement of l_t; and C_{p,p_t} is the cost of transitioning from position p_t to p (either 0.5 or 0). V_s is initialized at $R_{p,l}$. We used synchronous value iteration until the value matrix converged.

3.3. Results

Our test set contained 751,680 observations. Using the converged value matrix as a guide for actions, the Markov model generated a theoretical gross profit of \$110,900 on 1730 trades. Since each trade cost \$50, this amounted to a net profit of \$24,500, or \$2,816/day. The net profit per trade was \$14.

We also tried expanding the model to include the depth immediately below the best bid and above the best ask. This added 2 additional dimensions to the state space. To keep the problem tractable, these two additional dimensions were discretized into two buckets: less than the depth below/above it, or greater than the depth below/above it. This resulted in a state space of 1296 states, although only 432 were used in the transition matrix. With this expanded state space, the net profit was slightly higher at \$25,400 on 1806 trades.

4. Conclusion and Future Work

There are many factors that this model does not consider, such as execution delays, market impact, and transaction costs other than the spread. It is unlikely that this model would generate such substantial profits in practice. In particular, it would be better to model the movement in bid price and ask price separately, rather than just modeling movement in the mid-quote. Since trades for this strategy would happen at the bid and the ask, this would give more accurate estimates of the profitability of the system.

There is also room to improve the predictive power of the model. In particular, our final model discretized the last two dimensions into only two buckets in order to keep the transition matrix tractable. An expanded state space would probably result in a significant improvement.

Nevertheless, in this project we successfully demonstrated the effectiveness of adding depth to a model. It is clear that real-time information on depth can be used as an indicator of future short-term price movements.

[4]The spread is the difference between the bid price and the ask price, and is always non-zero. Thus, if one were to simultaneously buy and sell in the market, there would be a net loss equal to the spread.