Social Predict - Weighted Probability Adjustment Model (WPAM) for Market Pricing and Divergence-Based Payout Model (DBPM) for Payout Distributions

Patrick Delaney

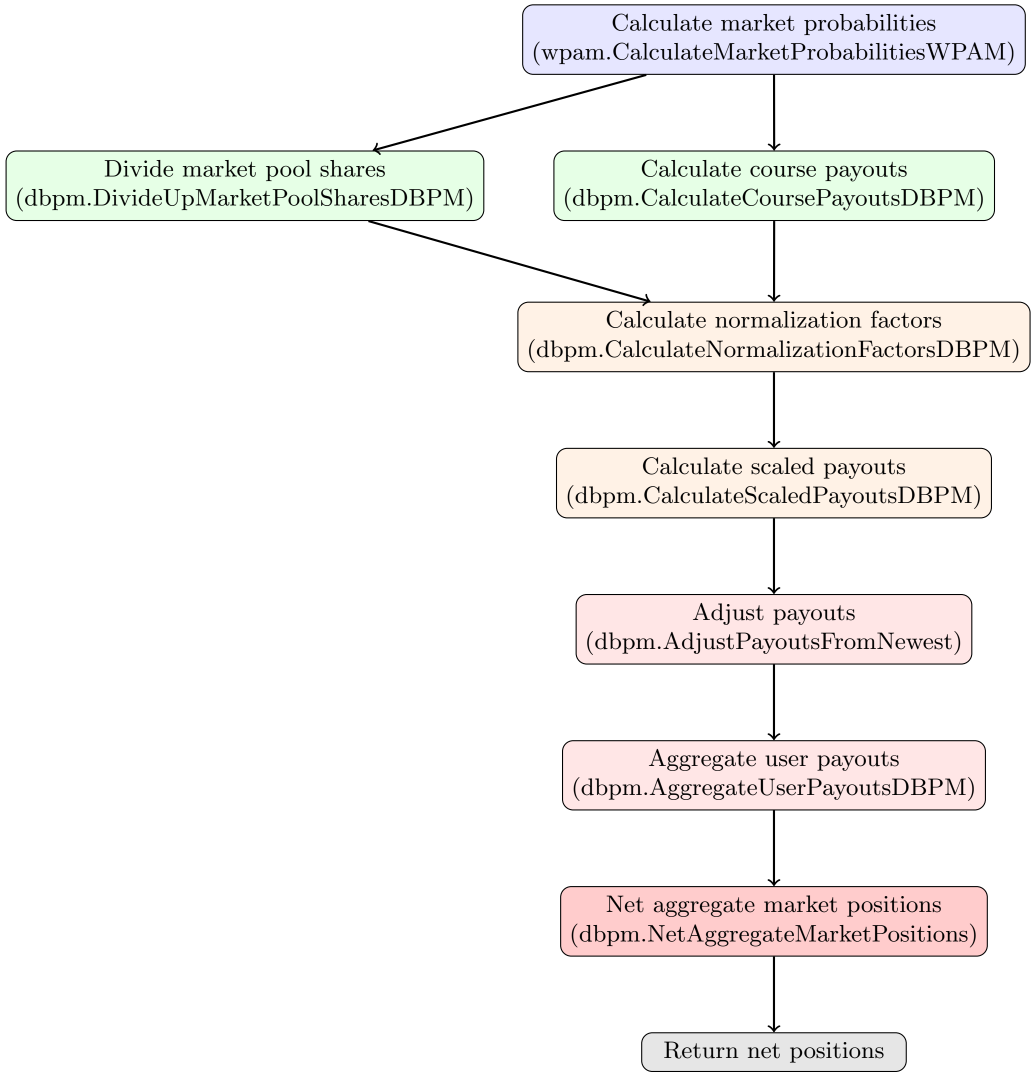
November 2024

1 Overview

- **Overall Purpose:** The purpose of this document is to describe in clear detail with graphical illustrations how market pricing (probability) gets set and how points/money/units gets distributed from a series of historical bets/transactions.
- **Ab Initio:** Since our software is stateless, the market price and how a pool of points gets distributed should be able to be calculated from the starting point of a string of transactions. There is no need to cache or store calculations along the way.
- **Transparency and Unit Testing:** Given that pricing and distribution is fundamental to how the software works, it is essential to ensure the full algorithm is well understood in order to write proper unit tests and to communicate the rules to users and participants.

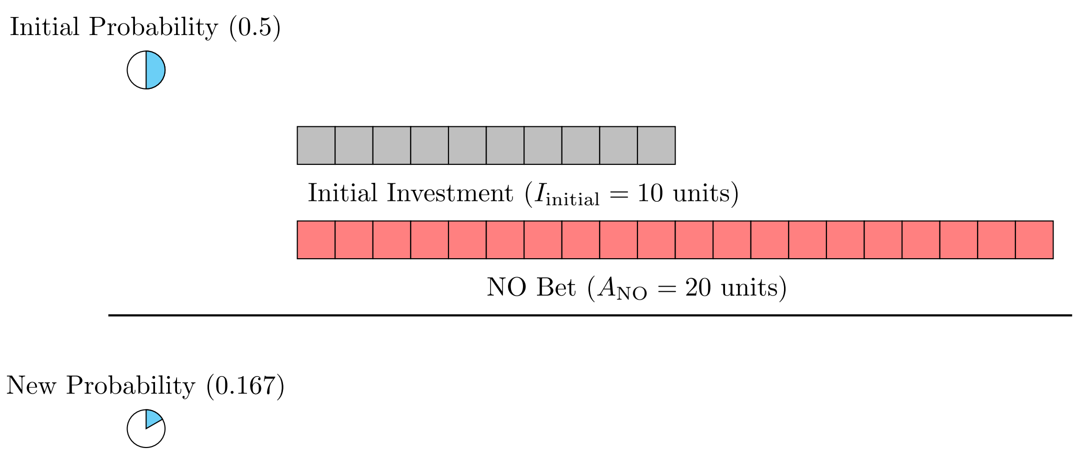
2 Positions Flow Cart

- **Exchange, Transform, Load:** The net positions that every trader has on any given market are calculated through what is basically an Exchange, Transform and Load (ETL) pipeline. We Exchange the data in the form of transactions, Transform it through a series of steps as shown in the chart below, and finally we Load the net positions as an object.
- **Works Anytime:** The ETL pipeline shown in the chart works at any point in the market's history, including after the resolution. Positions could be calculated for a market resolution at ay given percentage between 0 pc and 100 pc or at 0 (NO) or 100 (YES) pc resolutions.


Calculate market probabilities
(wpam.CalculateMarketProbabilitiesWPAM)
Divide market pool shares
(dbpm.DivideUpMarketPoolSharesDBPM)
Calculate course payouts
(dbpm.CalculateCoursePayoutsDBPM)
Calculate normalization factors
(dbpm.CalculateNormalizationFactorsDBPM)
Calculate scaled payouts
(dbpm.CalculateScaledPayoutsDBPM)
Adjust payouts
(dbpm.AdjustPayoutsFromNewest)
Aggregate user payouts
(dbpm.AggregateUserPayoutsDBPM)
Net aggregate market positions
(dbpm.NetAggregateMarketPositions)
Return net positions

3 First Transaction

- **Initial Investment:** Every market is assumed to have an initial investment of at least 1. This investment comes from a fee assessed to the market creator. This initial investment goes both in the numerator and denominator of the probability calculation.
- **Initial Probability:** Every market starts out at a probability of 0.5 which means even odds between a YES and NO outcome.
- **First Transaction:** In our sample below, the first transaction is a bet of 20 Units in the NO direction.



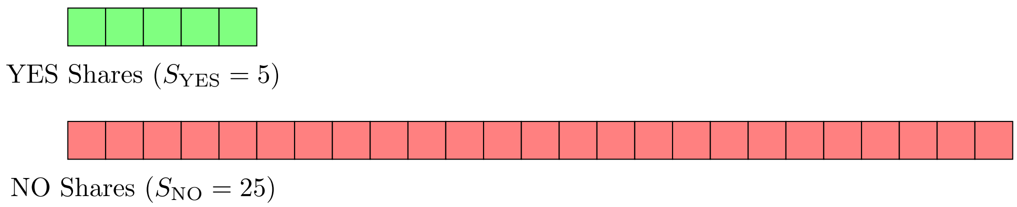
$$P_{\text{new}} = \frac{P_{\text{initial}} \times I_{\text{initial}} + A_{\text{YES}}}{I_{\text{initial}} + A_{\text{YES}} + A_{\text{NO}}} = \frac{0.5 \times 10 + 0}{10 + 0 + 20} = \frac{5}{30} \approx 0.167$$

Market Share Division

- **Banker's Rounded Share Pool:** First we look at how many total shares of YES and NO exist in the market pool as a whole by using the WPAM. The total share pool is equal to the Total Market Volume. The total share pool times the Probability will be the YES shares, while 1-(Probability) times the total share pool will be the NO shares.

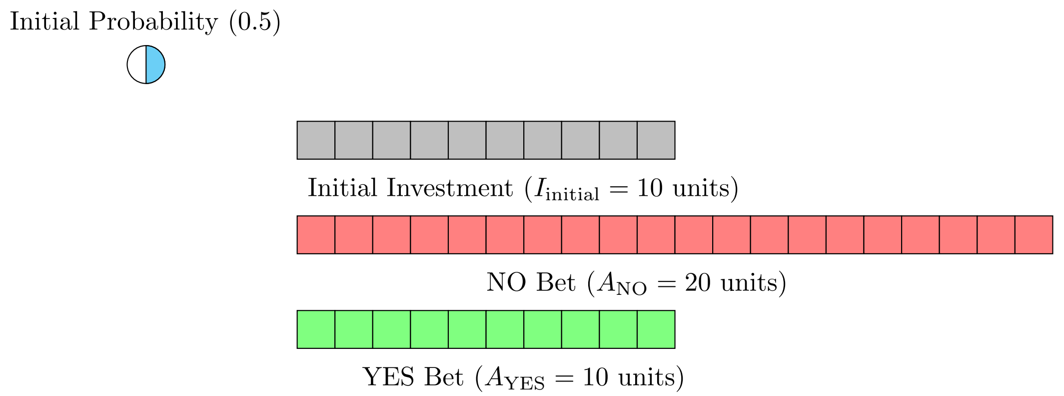
$$S = \text{Total Market Volume} + I_{\text{initial}} = 20 + 10 = 30 \text{ units}$$

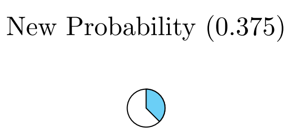
$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 30 \times 0.167 \rceil = 5$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 30 \times 0.833 \rceil = 25$$


YES Shares ($S_{\text{YES}} = 5$)
NO Shares ($S_{\text{NO}} = 25$)

4 Second Transaction

- **Second Transaction:** A different bettor comes in and places 10 units in the YES direction.



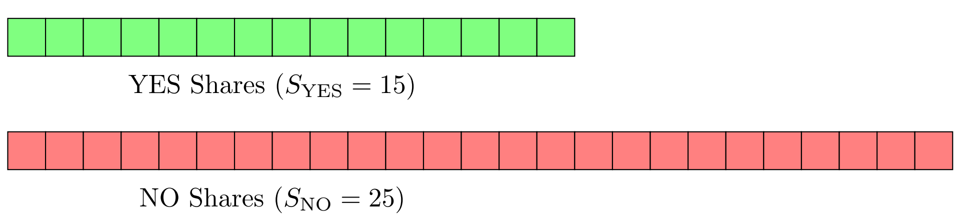


$$P_{\text{new}} = \frac{0.5 \times 10 + 10}{10 + 10 + 20} = \frac{15}{40} \approx 0.375$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (10 + 20) + 10 = 40 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 40 \times 0.375 \rceil = 15$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 40 \times 0.625 \rceil = 25$$



Calculating Course Payouts

- **Course Payouts:** The purpose of the Course Payouts is to create a relationship between the probability, which is a float, rational positive numbers, and shares, which are of type int64, integers. We use a linear weight for each historical bet in the history of the market to create a reward factor.
- **Reward Factor:** Course payouts are calculated in two steps, first by calculating a reward factor d_i, which is the distance from the current probability of the market R, and the probability at which the bet was made p_i.
- **New Probability After Bet, Not Previous Probability:** The current probability on the market, R is the New Probability which was just calculated by our WPAM function, $wpam.CalculateMarketProbabilitiesWPAM$ given the bet that was just immediately made. This enforces the idea that there is no reward for simply moving the market oneself and then selling at a new probability created from that movement. In order to profit, someone else must come in and move the market further in one's favor to be able to sell.

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations

Given:

- Resolution Probability: $R = 0.375$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$

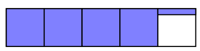
For Bet 1:

$$d_1 = |R - p_1| = |0.375 - 0.167| = 0.208$$
$$C_1 = d_1 \times b_1 = 0.208 \times 20 = 4.160 \text{ units}$$

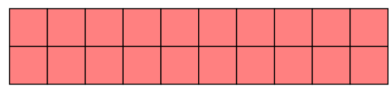
For Bet 2:

$$d_2 = |R - p_2| = |0.375 - 0.375| = 0$$
$$C_2 = d_2 \times b_2 = 0 \times 10 = 0 \text{ units}$$

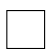
Course Payout for Bet 1 ($C_1 = 4.16$ units)



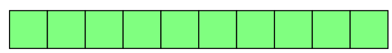
Bet 1: NO ($b_1 = 10$ units at $p_1 = 0.167$)



Course Payout for Bet 2 ($C_2 = 0$ units)



Bet 2: YES ($b_2 = 10$ units at $p_2 = 0.375$)



Normalization Factor Calculation

The normalization factor ensures that course payouts are proportional to the total shares for each outcome, making sure that payouts align with the available market pool.

- The normalization factor F_{YES} and F_{NO} adjusts each outcome's payouts so that they fit within the total shares allocated for each outcome.
- This prevents payouts from exceeding the total market pool, ensuring fair distribution based on the market's share allocation.
- By scaling payouts, the normalization factor maintains balance between YES and NO outcomes, reflecting the distribution of bets and probabilities in the market.

Step Three: Calculate Normalization Factor

- At this point the normalization factor will weight everything toward NO because there are zero course payouts for YES.

Given:

- Total YES Shares: $S_{\text{YES}} = 15$
- Total NO Shares: $S_{\text{NO}} = 25$
- Course Payouts:
  - $C_{\text{NO},1} = 4.16$ units
  - $C_{\text{YES},2} = 0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 0$, $C_{\text{NO\_SUM}} = 4.16$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 0$, we have:

$$F_{\text{YES}} = 0$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 4.16$, we have:

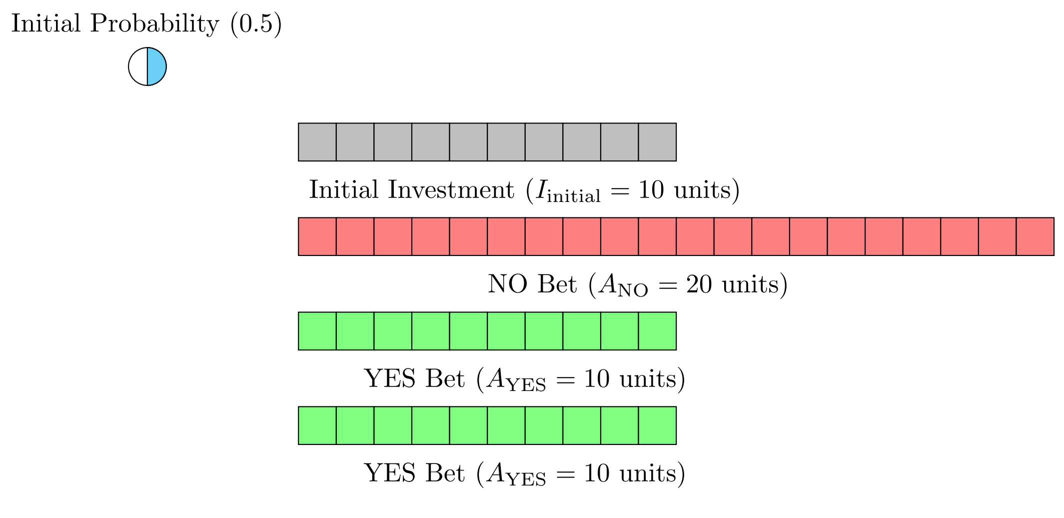
$$F_{\text{NO}} = \frac{25}{4.16} \approx 6.01$$

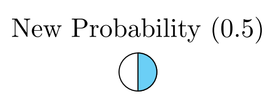
Thus, the normalization factors are:

$$F_{\text{YES}} = 0, \quad F_{\text{NO}} \approx 6.01$$

5 Third Transaction

- **Third Transaction:** Another bettor places 10 units in the YES direction.



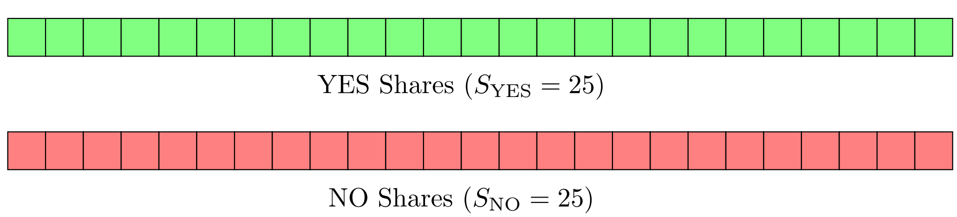


$$P_{\text{new}} = \frac{0.5 \times 10 + 20}{10 + 20 + 20} = \frac{25}{50} = 0.5$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (20 + 20) + 10 = 50 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 50 \times 0.5 \rceil = 25$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 50 \times 0.5 \rceil = 25$$



Calculating Course Payouts

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations for Third Transaction

Given:

- Resolution Probability: $R = 0.5$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$
- Bet 3: YES, $b_3 = 10$ units, $p_3 = 0.5$

For Bet 1:

$$d_1 = |R - p_1| = |0.5 - 0.167| = 0.333$$
$$C_1 = d_1 \times b_1 = 0.333 \times 20 = 6.660 \text{ units}$$

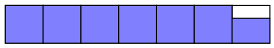
For Bet 2:

$$d_2 = |R - p_2| = |0.5 - 0.375| = 0.125$$
$$C_2 = d_2 \times b_2 = 0.125 \times 10 = 1.250 \text{ units}$$

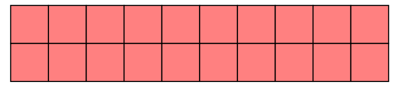
For Bet 3:

$$d_3 = |R - p_3| = |0.5 - 0.5| = 0$$
$$C_3 = d_3 \times b_3 = 0 \times 10 = 0 \text{ units}$$

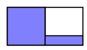
Course Payout for Bet 1 ($C_1 = 6.660$ units)



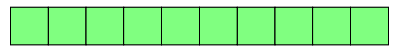
Bet 1: NO ($b_1 = 20$ units at $p_1 = 0.167$)



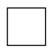
Course Payout for Bet 2 ($C_1 = 1.25$ units)



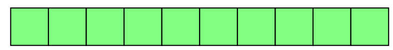
Bet 2: YES ($b_1 = 10$ units at $p_1 = 0.375$)



Course Payout for Bet 3 ($C_1 = 0.0$ units)



Bet 3: YES ($b_1 = 10$ units at $p_1 = 0.5$)



Normalization Factor Calculation for Third Transaction

Step Three: Calculate Normalization Factor

Given:

- Total YES Shares: $S_{\text{YES}} = 25$
- Total NO Shares: $S_{\text{NO}} = 25$
- Course Payouts:
  - $C_{\text{NO},1} = 6.66$ units
  - $C_{\text{YES},1} = 1.25$ units
  - $C_{\text{YES},2} = 0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 1.25$, $C_{\text{NO\_SUM}} = 6.66$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 1.25$, we have:

$$F_{\text{YES}} = \frac{25}{1.25} = 20$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 6.66$, we have:

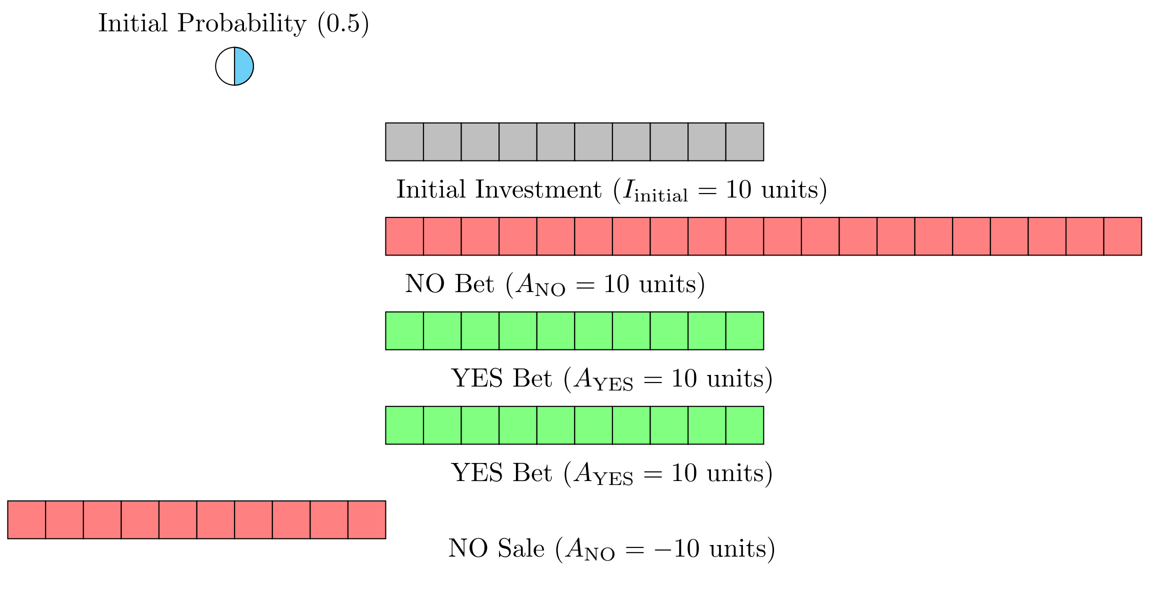
$$F_{\text{NO}} = \frac{25}{6.66} \approx 3.75$$

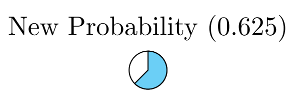
Thus, the normalization factors are:

$$F_{\text{YES}} = 20, \quad F_{\text{NO}} \approx 3.75$$

6 Fourth Transaction

- **Fourth Transaction:** The original bettor holding NO sells 10 units of NO.



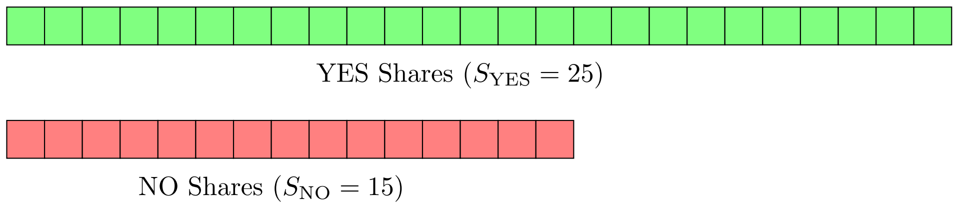


$$P_{\text{new}} = \frac{0.5 \times 10 + 20}{10 + 20 + 10} = \frac{25}{40} = 0.625$$

Market Share Division

$$S = \text{Total Market Volume} + I_{\text{initial}} = (20 + 10) + 10 = 40 \text{ units}$$

$$S_{\text{YES}} = \lfloor S \times P_{\text{new}} \rceil = \lfloor 40 \times 0.625 \rceil = 25$$
$$S_{\text{NO}} = \lfloor S \times (1 - P_{\text{new}}) \rceil = \lfloor 40 \times 0.375 \rceil = 15$$



Calculating Course Payouts

Step One: Calculate Reward Factor for Each Bet

$$d_i = |R - p_i|$$

Step Two: Calculate Course Payout for Each Bet

$$C_i = d_i \times b_i$$

Example Calculations for Fourth Transaction

Given:

- Resolution Probability: $R = 0.625$
- Bet 0: Neutral, $b_0 = 10$ units, $p_0 = 0.5$
- Bet 1: NO, $b_1 = 20$ units, $p_1 = 0.167$
- Bet 2: YES, $b_2 = 10$ units, $p_2 = 0.375$
- Bet 3: YES, $b_3 = 10$ units, $p_3 = 0.5$
- Bet 4: NO Sale, $b_4 = -10$ units, $p_4 = 0.625$

For Bet 1:

$$d_1 = |R - p_1| = |0.625 - 0.167| = 0.458$$
$$C_1 = d_1 \times b_1 = 0.458 \times 20 = 9.16 \text{ units}$$

For Bet 2:

$$d_2 = |R - p_2| = |0.625 - 0.375| = 0.25$$
$$C_2 = d_2 \times b_2 = 0.25 \times 10 = 2.5 \text{ units}$$

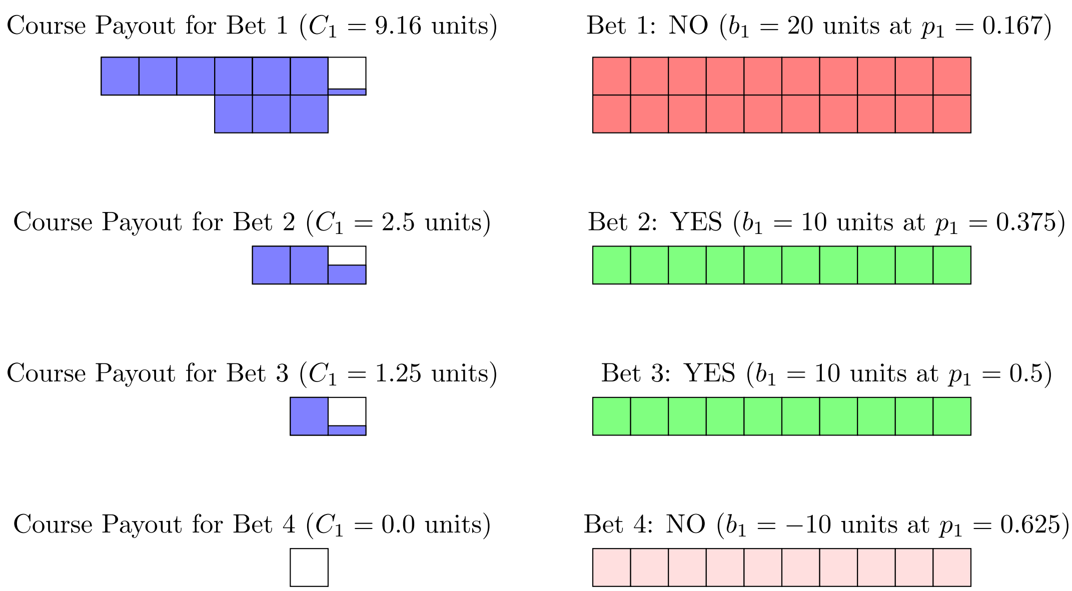
For Bet 3:

$$d_3 = |R - p_3| = |0.625 - 0.5| = 0.125$$

$$C_3 = d_3 \times b_3 = 0.125 \times 10 = 1.25 \text{ units}$$

For Bet 4:

$$d_4 = |R - p_4| = |0.625 - 0.625| = 0$$
$$C_4 = d_4 \times b_4 = 0 \times (-10) = 0 \text{ units}$$



Normalization Factor Calculation for Fourth Transaction

Step Three: Calculate Normalization Factor

Given:

- Total YES Shares: $S_{\text{YES}} = 25$
- Total NO Shares: $S_{\text{NO}} = 15$
- Course Payouts:
  - $C_{\text{YES},1} = 2.5$ units
  - $C_{\text{NO},1} = 9.16$ units
  - $C_{\text{YES},2} = 1.25$ units
  - $C_{\text{NO},2} = 0.0$ units
  - Total payout sums: $C_{\text{YES\_SUM}} = 3.75$, $C_{\text{NO\_SUM}} = 9.16$

Calculate YES Normalization Factor

$$F_{\text{YES}} = \begin{cases} \frac{S_{\text{YES}}}{C_{\text{YES\_SUM}}} & \text{if } C_{\text{YES\_SUM}} > 0 \\ 0 & \text{if } C_{\text{YES\_SUM}} = 0 \end{cases}$$

Since $C_{\text{YES\_SUM}} = 3.75$, we have:

$$F_{\text{YES}} = \frac{25}{3.75} = 6.67$$

Calculate NO Normalization Factor

$$F_{\text{NO}} = \begin{cases} \frac{S_{\text{NO}}}{C_{\text{NO\_SUM}}} & \text{if } C_{\text{NO\_SUM}} > 0 \\ 0 & \text{if } C_{\text{NO\_SUM}} = 0 \end{cases}$$

Since $C_{\text{NO\_SUM}} = 9.16$, we have:

$$F_{\text{NO}} = \frac{15}{9.16} \approx 1.64$$

Thus, the normalization factors are:

$$F_{\text{YES}} \approx 6.67, \quad F_{\text{NO}} \approx 1.64$$