On Market Makers Performance

Luke Camery
Advisor Amy Greenwald
Enrique Areyan Viqueira

May 12, 2017

1 Motivation

The Logarithmic Market Scoring Rule (LMSR), while popular in the theoretical literature, faces several critical barriers to adoption. The first is that the **market maker** is bounded loss, which is a non-starter for most real world applications. The second is that the **market maker** is not sensitive to liquidity. This means that a improperly configured **market maker** might allow any agent with a positive budget to drastically swing the price or at the other extreme, not allow any agent to modify the price much at all. LMSR is well studied, however, because it has Path Independence and Translation Invariance. Many other theoretical **market makers** have been proposed that also have these two properties, but with additional benefits, and some offer tradeoffs in order to gain more desirable properties. This work seeks to quantify these tradeoffs empirically.

2 Goals

To test three different market maker mechanisms, including a novel design, for their lprofit expectation and accuracy.

3 Model

3.1 Definitions

We denote **time** by $\tau \in \mathbb{R}_+$.

An **event** e has an outcome at time $e_t \in \mathbb{R}_+$. Observe that we restrict our attention to binary outcomes. The outcome is equal to YES in case the event occurs, and NO otherwise. We denote the YES outcome with a 1, and the NO outcome with a 0. We assume there is a way to unambiguously determine the outcome of an event.

R is an **oracle** for mapping an event to an outcome. We denote the outcome of event e under R as $R(e) \in \{0, 1\}$.

An **option** $o = \langle o_e, o_d \rangle$ is a security that yields a return depending on the outcome of an event o_e. Each option has a direction $o_d \in \{0, 1\}$. The option will convert to \$1 at time e_t if $R(o_e)$ equals o_d. Otherwise, it converts to \$0. Two options are said to be **complementary** if they represent opposite directions on the same event.

An **agent** $a \in A$ has a **private belief** $a_v \in 0, 1$ and a **budget** $a_b \in \mathbb{R}_+$ about event e. The agent's private belief a_v is the subjective probability that the agent assigns to the outcome o_e being direction o_d at strike time e_t.

A **prediction market** M is a forum for trading complementary options. Formally, a prediction market is a tuple $M = \langle o_0, o_1, A, q_0, q_1 \rangle$ where each **agent** $a \in A$ purchases either some number of o_0 or o_1 options paying the price quoted by the market maker. We denote the total quantity purchased of o_0 and o_1 as q_0 and q_1 respectively. These quantities are always initialized to zero.

A **market maker** K provides liquidity in a market M by offering to trade either o_0 or o_1 at a price p that it determines. Formally, K is a function that maps an option o, and a quantity $q \in \mathbb{R}_+$ at time $\tau \in [0, t)$ to a price $p(o, q, \tau) \in \mathbb{R}_+$. By definition, $p(o, q, \tau) = R(o_e)$, if $\tau \geq t$.

3.2 Assumptions

In our model we assume without loss of generality that agent $a \in A$ has an arrival time $a_\tau \in \mathbb{R}_+$ where $a_\tau < t$, and execute its strategy only once at time a_τ.

We assume that agents' private beliefs and budgets are drawn from distributions, which are known, $v \sim F(\cdot)$ and $b \sim F(\cdot)$.

We assume agents to observe the current price of both option o_0 or o_1.

We assume that the market maker K and all its parameters are common knowledge.

4 Equilibria

We will use two classic equilibria concepts from the literature, and a third of our design. In both equilibria, agents are assumed to be risk neutral utility maximizers.

4.1 REE and PI

The two classic equilibria concepts that we will utilize are the **Rational Expectations Equilibrium** (REE) and the **Prior Information Equilibrium** (PI). Rational Expectations hypothesizes that all agents act as if they had the collective belief. The collective belief aggregates the individual beliefs received by each agent. This implies that the prediction markets should be as accurate as the collective belief. Under our model, this implies that prediction market's final belief will match the aggregate belief.

Prior Information hypothesizes that all agents act on a linear combination of their private information and the current market price as Bayesian updaters. PI implies that agents will take into account the current market price and their own belief. Agents are willing to participate at their expected value given the two beliefs.

A major shortcoming in the current literature, which is reflected in the agent behavior theories, is the generalization that agents cannot choose when to enter the market (Hanson, 2003). Although fixing entry time simplifies equilibrium calculations, in practice agents strategize about the value of a certain entry time τ. Under a model based in REE where the collective belief is truthful, a rational agent would want to be the last decision maker and then have the outcome revealed with certainty.

4.2 No Trade Theorem

Under relaxed assumptions from REE and PI, agents wait until $x \in [0, |A|]$ agents have executed their strategies depending on their valuation of the information gained by waiting. It is trivial to show that the valuations are monotonically increasing with x. Each spot $\forall x$ is worth $\frac{x}{|A|} E\left[o\right]$ since each increment in x provides an equal increase in information. This implies that rational agents would need to be paid the difference between their riskless profits and the value of x in order to select spot $x|E\left[o\right](1 - \frac{x}{|A|})$. In the absence of this payment, it is individually rational for agents to wait until the first half of agents have already entered the market before trading. Therefore, lacking any external payments, no agent will enter the market and no trades will occur.

4.3 Agent Strategy

In order to assess accuracy in situations where agents have variable information, we will consider different types of agent strategies in our experiments. There are two types of agent strategies **shortsighted** S and **farsighted** F. Similarly, **informed** I agents have an exogenous belief about the outcome whereas **uninformed** U agents can only base their decision on the market price.

The following agent strategies build off of the work on Kelly Agents, Constant Relative Risk Aversion Agents, (Kets et al, 2014) and Zero Intelligence Agents (Othman, 2008). Kelly and CRRA Agents are limiting because they introduce risk aversion in order to limit the bets that agents place. This is empirically less accurate since online betting environments are known to attract risk seeking, or at least risk neutral, traders. Zero Intelligence Agents inspire our Shortsighted class of agents since they are short sighted actors. Kelly and CRRA Agents inspire our Uninformed class of agents since their purpose is to measure the persistence of inaccurate traders. We believe that **Shortsighted, Farsighted, and Un/Informed** agents cover a more abstract class of trader behavior. All four strategies are myopic.

The following agent strategies build off of the work on Kelly Agents, Constant Relative Risk Aversion Agents, (Kets et al, 2014) and Zero Intelligence Agents (Othman, 2008). Kelly and CRRA Agents are limiting because they introduce risk aversion in order to limit the bets that agents place. This is empirically less accurate since online betting environments are known to attract risk seeking, or at least risk neutral, traders. Zero Intelligence Agents inspire our Shortsighted class of agents since they are short sighted actors. Kelly and CRRA Agents inspire our Uninformed class of agents since their purpose is to measure the persistence of inaccurate traders. We believe that **Shortsighted, Farsighted, and Un/Informed** agents cover a more abstract class of trader behavior.

Prediction markets are known to be myopically incentive compatible, which means that myopic agents bid truthfully. There is much work demonstrating that this assumption fails when agents can use **bluffing** and **reticence** to mislead other agents and profit off of that deception. We will show that when restricting the game to a single shot for each agent and using market scoring rule based mechanisms that it is impossible to design a dominant non-myopic strategy.

4.4 Nonmyopic Lemma

Consider a non-myopic strategy that is dominant for an agent in a MSR based market. That agent has a final belief, which incorporates whatever logic and prior information the agent has, called a_v. The non-myopic strategy must either move the price p towards their belief a_v, but less than their budget a_b permits, or away from it. In the case where the non-myopic strategy moves the price away from their belief, the agent will be scored on how much their price movement improves the market prediction, which is mapped to the price. In order to make positive profit, the belief needs to improve. If the agent thinks that this non-myopic strategy is improving the price then they cannot hold belief a_v, which is a contradiction. In the second case, if the agent holds belief a_v then it would be strictly preferable to move the price as close as possible to a_v. This means that the myopic strategy strictly dominates the non-myopic strategy. Since in both cases the myopic strategy strictly dominates therefore the myopic strategy is dominant under this model.

5 Market Makers

In these experiments we will test the following three market markers M.

5.1 Logarithmic Market Scoring Rule

LMSR is a strictly proper scoring rule developed by Robert Hanson. A scoring rule is strictly proper if the forecaster has no incentive to report anything but their true belief. This condition is the same as stating that LMSR is myopically incentive compatible.

LMSR uses a logarithmic cost function:
$C(q_0, q_1) = b\ln(e^{\frac{q_0}{b}} + e^{\frac{q_1}{b}})$. The cost charged to a trader wanting to buy q_a shares of o_0 and q_b shares of o_1 is: $C(q_0 + q_a, q_1 + q_b) - C(q_1, q_1)$. Traders are charged for their movement in the market prediction. b is the liquidity parameter set ex ante by the market maker. It controls how much the market maker can lose and also adjusts how easily a trader can change the market price. The market maker always loses up to $b\ln(2)$. A large b means that it would cost a lot to move the market price while a small b makes large swings relatively inexpensive. Formally, LMSR is $M(b)$.

The instantaneous price of LMSR market M, which is also the market's prediction for the option $o_{\{0,1\}}$, is quoted with: $p_{\{0,1\}} = \frac{e^{\frac{q_{\{0,1\}}}{b}}}{e^{\frac{q_0}{b}} + e^{\frac{q_1}{b}}}$.

Advantages

1. Path Independence - any way the market moves from one state to another state yields the same payment or cost to the traders in aggregate [Hanson 2003]
2. Translation Invariance - all prices sum to unity. (Direct mapping to a probability.)

Disadvantages

1. Liquidity Insensitive - the market cannot adjust to periods with low or high activity. The market maker must set the liquidity parameter based on their prior belief, but has little to no guidance on how to set it.

2. Guaranteed Loss - the market maker cannot profit and has a guaranteed bounded loss.

5.2 LMSR Algorithms

Algorithm: cost

input : state $\langle q_0, q_1 \rangle$, liquidity parameter b, to trade q, **direction** $\{0, 1\}$
output: cost
oldScore $= b \exp\left(\frac{q_0}{b}\right) + \exp\left(\frac{q_1}{b}\right)$
if *direction == YES* **then**
  newScore $= b \exp\left(\frac{q_0+q}{b}\right) + \exp\left(\frac{q_1}{b}\right)$
else
  newScore $= b \exp\left(\frac{q_0}{b}\right) + \exp\left(\frac{q_1+q}{b}\right)$
return newScore - oldScore

Algorithm: price

input : state $\langle q_0, q_1 \rangle$, liquidity parameter b
output: price
quantity, price = getQuantityPrice(state, b,null,true)
return price

Algorithm: quantity

input : state $\langle q_0, q_1 \rangle$, liquidity parameter b, belief a_v, direction $\in \{\text{true}, \text{false}\}$
output: quantity q
quantity, price = getQuantityPrice(state,b,belief,direction)
return quantity

Algorithm: getQuantityPrice

input : state $\langle q_0, q_1 \rangle$, liquidity parameter b, belief a_v, direction $\in \{\text{true}, \text{false}\}$
output: quantity q, price p
if *belief == null* **then**
  quantity = 0
else
  if *direction* **then**
    price = belief
    side = q_0
    top = q_1
  else
    price = 1 - belief
    side = q_1
    top = q_0
  quantity $= b \log\left(\frac{\text{price} \exp\left(\frac{\text{top}}{b}\right)}{1-\text{price}}\right) - \text{side}$
if *direction* **then**
  newPrice = price(q_0 + quantity, q_1, b)
else
  newPrice = price(q_0, q_1 + quantity, b)
return (quantity, newPrice)

Algorithm: capitalToShares

input : market maker $K = \langle q_0, q_1, b \rangle$, money m, direction $\in \{\text{true}, \text{false}\}$
output: quantity
if *direction* **then**
| side = q_0 top = q_1
else
| side = q_1 top = q_0
return $b \log \left(\exp \left(\frac{m}{b} + \log \left(\exp \left(\frac{M_{q_0}}{b} \right) + \exp \left(\frac{M_{q_1}}{b} \right) \right) \right) - \exp \left(\frac{\text{top}}{b} \right) \right) - \text{side};$

5.3 Practical Liquidity Sensitive Market Maker

The Liquidity Sensitive Market Maker uses the underlying LMSR mechanism but invokes a novel function for setting the liquidity parameter b. The function is: $b(q) = \alpha \sum_i q_i$, where q represents the quantity of each option on the market. Binary prediction markets, which we have restricted ourselves to, reduce the function to $b(q) = \alpha(q_0 + q_1)$. The parameter α between $(0, 1]$ that represents the commission the market maker skims off each transaction. A larger α yields a higher commission.

The LSMM uses this formula for setting b to make the market maker profitable and to make it liquidity sensitive, meaning that the market maker charges traders differently depending on the market depth.

Advantages

1. Path Independence - any way the market moves from one state to another state yields the same payment or cost to the traders in aggregate [Hanson 2003]
2. Liquidity Sensitive - the market adjusts to periods with low or high activity. The market maker decreasingly subsidizes the market as activity rises.
3. Guaranteed Profit - the market maker has unbounded profit but bounded loss at near 0.

Disadvantages

1. Translation Variance - all prices sum beyond unity. (No direct mapping to a probability though it does provide a tight range.)
2. Market makers are incentivized to raise their commission to 1, which not only hurts traders, but also increases the valid probability range and decreases the number of traders who are willing to trade with the market maker. See $\frac{1}{n} - \alpha(n-1)\ln(n) \leq p(q_i) \leq \frac{1}{n} + \alpha \ln(n)$.

5.4 LSMM Algorithms

Algorithm: $b(\alpha)$

input : alpha α
output: liquidity parameter b
return $\alpha(M_{q_0} + M_{q_1})$;

5.5 Luke's New MM

The current function is $b(q) = \alpha \left[\sum_i (q_i) + t\right]$ where t represents the number of transactions that have occurred.

6 Strategies

An agent's **strategy** $s_a(\tau) \in \mathbb{R}_+$ specifies the quantity of option $o_{0,1}$ purchased by the agent at time τ, $\forall \tau \leq e_t$

$s_a^{S^I}(\tau)$ is an **informed agent** holding exogenous belief a_v who is willing to pay up to a_b in order to move the market price M_p as close as possible to their belief a_v.

Algorithm: Shortsighted Informed Strategy S^I

Data: belief a_v, market marker K
Result: $M_{p,\tau} \rightarrow M_{p,\tau+1}$
initialization
if $M_p < a_v$ **then**
  maxShares = capitalToShares(K, a_b, YES)
  wantedShares = priceToShares(K, a_v, YES)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of YES from K
else
  maxShares = capitalToShares(K, a_b, NO)
  wantedShares = priceToShares(K, a_v, NO)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of NO from K

$s_a^{S^U}(\tau)$ is an **uninformed agent** holding no exogenous belief a_v who who is willing to pay up to a_b in order to move the market price p_M as close as possible to 1 if at time τ, $p_{M\tau} \geq .5$ otherwise 0.

Algorithm: Shortsighted Uninformed Strategy S^U

Data: belief a_v, market M, market maker K
Result: $M_{p,\tau} \rightarrow M_{p,\tau+1}$
initialization
if $M_p > .5$ **then**
  maxShares = capitalToShares(K, a_b, YES)
  wantedShares = priceToShares(K, 1, YES)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of YES from K
else
  maxShares = capitalToShares(K, a_b, NO)
  wantedShares = priceToShares(K, 0, NO)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of NO from K

$s_a^{F^I}(\tau)$ is an **informed agent**holding exogenous belief a_v who is attempting to maximize the expected value by bidding based on a linear combination of their belief a_v and the current market price p_M accounting for how many agents x have already bid.

Algorithm: Shortsighted Informed Strategy F^I

Data: belief a_v, market marker K
Result: $M_{p,\tau} \to M_{p,\tau+1}$
initialization
linearBelief = $\frac{\tau * M_p + a_v}{\tau+1}$
if $M_p < a_v$ **then**
  maxShares = capitalToShares(K, a_b, YES)
  wantedShares = priceToShares(K, linearBelief, YES)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of YES from K
else
  maxShares = capitalToShares(K, a_b, NO)
  wantedShares = priceToShares(K, 1-linearBelief, NO)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of NO from K

$s_a^{F^U}(\tau)$ is an **uninformed agent** holding exogenous belief a_v who is attempting to maximize the expected value by bidding based on the current market price p_M accounting for how many agents x have already bid.

Algorithm: Shortsighted Informed Strategy F^U

Data: belief a_v, market marker K
Result: $M_{p,\tau} \to M_{p,\tau+1}$
initialization
linearBelief = $\frac{\tau * M_p + 1}{\tau+1}$
if $M_p > .5$ **then**
  maxShares = capitalToShares(K, a_b, YES)
  wantedShares = priceToShares(K, linearBelief, YES)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of YES from K
else
  maxShares = capitalToShares(K, a_b, NO)
  wantedShares = priceToShares(K, 1-linearBelief, NO)
  sharesDemanded = min(maxShares, wantedShares)
  buy sharesDemanded of NO from K

---

7 Benchmarks

We establish two benchmarks for the market making mechanisms. The first is market maker profit. This is essential because LMSR and our trade incentivizing market makers often operate at no profit or at a loss. The second is accuracy. We will evaluate accuracy according to expected price as determined by our three equilibrium concepts.

7.1 Profit

Definition 7.1. (Market Maker Revenue) Given a market maker p_M and a set of participating agents A_o, the revenue obtained from p_M is defined as

$$R(p_M, A_o) = \sum_{a \in A_o} \left[\int_{\tau=0}^{\tau=o_\tau} p_M(o_0, a, \tau, s_a^0(\tau)) s_a^0(\tau) dt + \int_{\tau=0}^{\tau=o_\tau} p_M(o_1, a, \tau, s_a^1(\tau)) s_a^1(\tau) dt \right]$$

Definition 7.2. (Market Maker Cost) Given a market maker p_M and a set of participating agents A_o, the cost to p_M is defined as

$$C(p_M, A_o) = \sum_{a \in A_o} \left[\int_{\tau=0}^{\tau=o_\tau} R(o_e) s_a^0(\tau) dt + \int_{\tau=0}^{\tau=o_\tau} R(o_e) s_a^1(\tau) dt \right]$$

Definition 7.3. (Market Maker Proft) Given a market maker p_M and a set of participating agents A_o, the profit of p_M is defined as

$$P(p_M, A_o) = R(p_M, A_o) - C(p_M, A_o)$$

Definition 7.4. (Profit-Maximizing Market Maker). Among all Market Makers L, given a set of participating agents A_o find the one that maximizes Profit:

$$PM(L, A_o) = argmax_{a \in A_o} P(p_M, A_o)$$

7.2 Accuracy

We will use the following definitions of accuracy from our equilibrium concepts.

Rational Expectations and **Prior Information** provide three definitions for ground truth against which we can compare the final prediction for our **market makers**. All three equilibria suggest that the market's final prediction will predict the true outcome.

Rational Expectations implies that the final price will be the average of each agent's signal since all agents have equal budgets. As we have restricted signals to the set of $0, 1$, the expected price will simply be the number of 1 signals divided by the total number of agents $\frac{|A_1|}{|A|}$. Given that the aggregate signal is truthful, this implies that the market is 100% accurate. For example, with a signal vector $[0, 1, 1, 0, 1]$, the expected price is .6 and the prediction will be correct.

Prior Information implies that the final price will be between the weighted average and .5 because it is biased towards the initial price. The prediction will still be accurate just with lower confidence. For example, with a signal vector $[0, 1, 1]$, the first agent will target .25 since it is the Bayesian update between their prior 0 and the price .5. The second agent will target .5 since it is the Bayesian update between their prior 1 and the price .25 giving .25 double weight. The third agent will target .625 since it is the Bayesian update between their prior 1 and the price .5 giving .5 triple weight. The outcome will still be correct at .625, but less than .66 which is the RE prediction in this case.

8 Experimental Design

For our experiments, we will assume that private beliefs are $0, 1$ and that budgets are equal.

8.1 Metrics

For each of the following configurations, we will rank our three market makers against our two benchmarks defined above: **Profit-Maximizing Market Maker** and **Accuracy** according to RE and PI.

8.2 Configurations

8.2.1 Market Makers

We will run each agent arrangement against the following market makers:

1. **Logarithmic Market Scoring Rule** with varying b values. Since there is little intuition behind setting b other than the loss incurred by the market maker, we want to evaluate a full range of b options that appear in other papers. Therefore we will use every integer value for b between 1 and 100 as those correspond to natural extremes where the price is trivially malleable to where it is near static.

2. **Liquidity Sensitive Market Maker** with varying alpha parameters. Since alpha corresponds to the commission, we will assess all α values on the interval [0,1] with two digits of precision. This evaluates the full range of logical values.

3. **Time Sensitive Market Maker** with varying α parameters. Since α corresponds to the commission, we will assess all α values on the interval [0,1] with two digits of precision. This evaluates the full range of logical values.

8.3 Agents

We will run each of the following configurations with 9 agents, 49 agents, and 99 agents. Prediction markets vary between miniscule and small in market depth so this represents a reasonable range for evaluation. The odd numbers prohibit market outcomes at .5. The final price for each configuration will be the average of the prices that result from every ordering of the agents.

1. Informed and Uninformed Shortsighted Agents with a majority to minority split, $\frac{2}{3}$ to $\frac{1}{3}$ split, and $\frac{9}{10}$ to $\frac{1}{10}$ split.

2. Informed and Uninformed Farsighted Agents with a majority to minority split, $\frac{2}{3}$ to $\frac{1}{3}$ split, and $\frac{9}{10}$ to $\frac{1}{10}$ split.

3. Informed Shortsighted and Informed Farsighted Agents with a majority to minority split, $\frac{2}{3}$ to $\frac{1}{3}$ split, and $\frac{9}{10}$ to $\frac{1}{10}$ split.

4. Informed Farsighted and Uninformed Shortsighted Agents with a majority to minority split, $\frac{2}{3}$ to $\frac{1}{3}$ split, and $\frac{9}{10}$ to $\frac{1}{10}$ split.

8.4 Configuration Table

Name	Market Maker	Agent Types	Agent Split
1	LMSR	S^I, S^U	5,4
2	LMSR	S^I, S^U	6,3
3	LMSR	S^I, S^U	8,1
4	LMSR	F^I, F^U	5,4
5	LMSR	F^I, F^U	6,3
6	LMSR	F^I, F^U	8,1
7	LMSR	S^I, F^I	5,4
8	LMSR	S^I, F^I	6,3
9	LMSR	S^I, F^I	8,1
10	LMSR	F^I, S^U	5,4
11	LMSR	F^I, S^U	6,3
12	LMSR	F^I, S^U	8,1

8.5 Simulator

Since LMSR based mechanisms provide constant time transactions, we can reliably converge each of these markets within a tightly bounded interval. Based on tests using our Java Virtual Machine based simulation, each agent takes a conservative upper bound of 15 milliseconds to fully enter the market. This means that the worst amount of time that a market can take in these experiments is 1.5 seconds.

9 Markov Decision Process

Since LMSR does not necessarily aggregate beliefs according to the REE with one preset b, we want to devise an algorithm for setting b such that we recover the REE aggregate belief. In addition, since b determines the market maker's loss, we want the selection of b values to be the least costly selection. Therefore we will solve the following Markov Decision Process to minimize cost while recovering the aggregate belief.

9.1 Definitions

A Markov Decision Process (MDP) $M = \langle \mathcal{S}, \mathcal{A}, T, R \rangle$ is defined by a set of states, an action space $\mathcal{A}$, a deterministic transition probability function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, and a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

9.2 Construction

In the following, we construct an MDP to model the cost of the paths of b's through the MDP that result in the final price equaling (close to) the equilibrium price P, which is derived from the Rational Expectations Equilibrium. We assume the order in which agents arrive to trade is known and fixed as $\langle a_0, a_1, ..., a_n \rangle$, where $a_i \in A$ and a_0 is a dummy agent with belief .5. We assume agents are truthfully reporting their beliefs a_i as LMSR incentivizes them to do.

Given an LMSR market maker, we construct the MDP as follows.

The set of states is $S = \{s_0, s_1, ..., s_n, \text{YES}, \text{NO}, \text{END}\}$, where $s_i = \langle q_0, q_1 \rangle$ for $q_0, q_1 \in \mathbb{R}_+$, respectively, the total quantity of YES securities and total quantity of NO securities sold by the market maker. There are two special states YES and NO. There is also an absorbing state END.

The set of actions is $A(s_i) = [b_0, b_1]$, where b_0 and b_1 represent the range of valid liquidity parameter values, determined by b. The set of actions A(YES) = A(NO) = A(END) = $\emptyset$.

We define $q_0' = q_0$ + getQuantity($\langle q_0, q_1 \rangle$, b, a_i, YES), and $q_1' = q_1$ + getQuantity($\langle q_0, q_1 \rangle$, b, a_i, NO).

Transitions are deterministic. When $i = 0, \ldots, n-1, T(s_i, b) = \langle q_0', q_1' \rangle$, where $s_i = \langle q_0, q_1 \rangle$. When $i = n$,

$$T(s_n) = \begin{cases} \text{YES} & \text{if R}(o) = 1 \\ \text{NO} & \text{if R}(o) = 0 \end{cases}$$

When $i = n+1$, $T(\text{YES}) = T(\text{NO}) = \text{END}$. As T is absorbing, $T(\text{END}) = \text{END}$.

Rewards are defined as follows: when $i = 0, \ldots, n$, $R(s_i, b) = \text{cost}(q_0', q_1') - \text{cost}(q_0, q_1)$, where $s_i = \langle q_0, q_1 \rangle$, and q_0', q_1' are defined above. In addition,

$$R(\text{YES}) = \begin{cases} -q_0 & \text{if getPrice}(\langle q_0, q_1 \rangle, b) = P \\ -\infty & \text{otherwise} \end{cases}$$

$$R(\text{NO}) = \begin{cases} -q_1 & \text{if getPrice}(\langle q_0, q_1 \rangle, b) = P \\ -\infty & \text{otherwise} \end{cases}$$

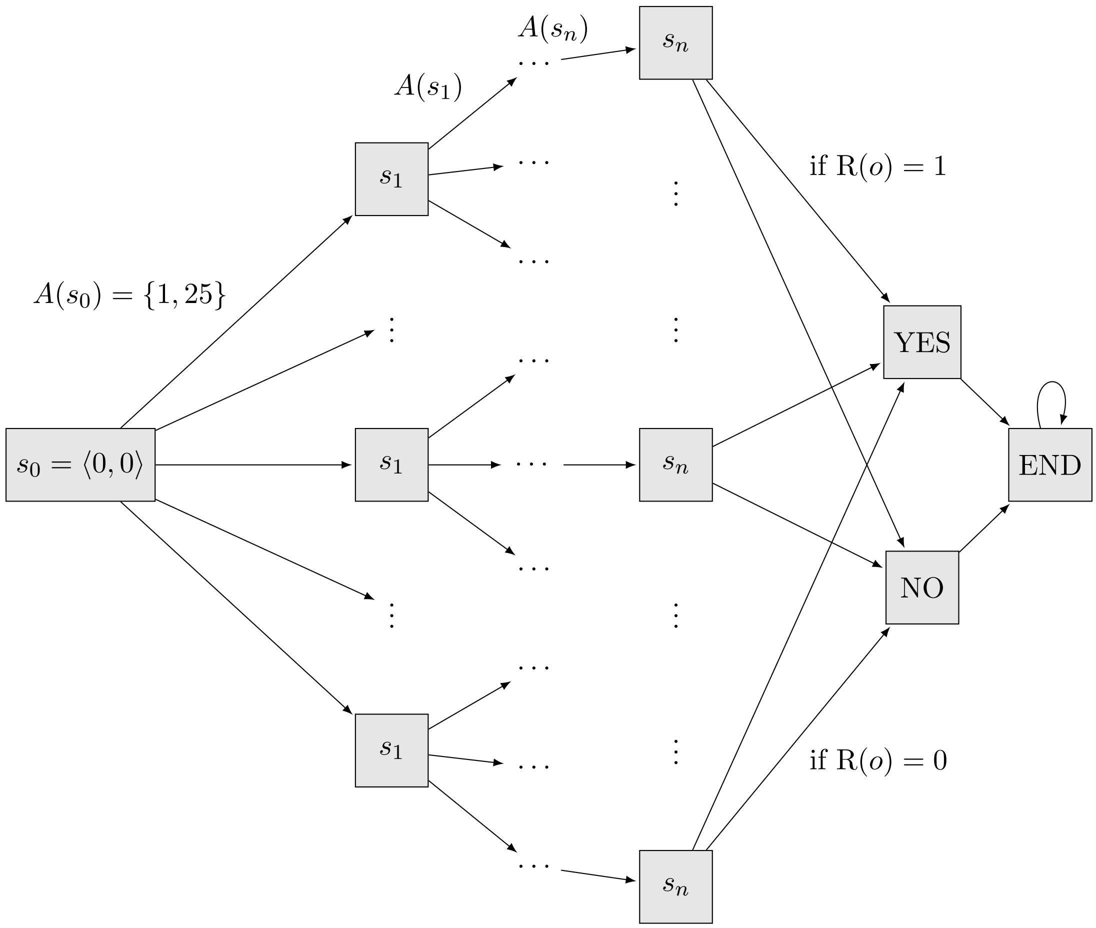
$$R_{n+1}(\text{END}) = \; 0 \; .$$

9.3 Assumptions

We will use a discretization factor of $\delta = 1$ for our action space of b values. Since in practice b is exhibits odd behavior below 1, we restrict our action space to $A = \{1, ...\}$, but we will only search over $A = \{1, 25\}$ for efficiency.

Due to finite precision, we will permit a tolerance of ϵ for the final market price such that $|P - \text{getPrice}(\langle q_0, q_1 \rangle, b)| < \epsilon$.

9.4 Diagram



10 Conclusion

10.1 Future Directions