

Article

On Laws of Thought—A Quantum-like Machine Learning Approach

Lizhi Xin¹, Kevin Xin² and Houwen Xin^{3,*}

¹ Independent Researcher, Hefei 230026, China; lizhi_xin@yahoo.com

² Independent Researcher, Chicago, IL 60607, USA; kevin_xin@yahoo.com

³ Department of Chemical Physics, University of Science and Technology of China, Hefei 230052, China

* Correspondence: hxin@ustc.edu.cn

Abstract: Incorporating insights from quantum theory, we propose a machine learning-based decision-making model, including a logic tree and a value tree; a genetic programming algorithm is applied to optimize both the logic tree and value tree. The logic tree and value tree together depict the entire decision-making process of a decision-maker. We applied this framework to the financial market, and a “machine economist” is developed to study a time series of the Dow Jones index. The “machine economist” will obtain a set of optimized strategies to maximize profits, and discover the efficient market hypothesis (random walk).

Keywords: quantum-like decision theory; genetic programming; machine learning; quantum gates; efficient market hypothesis

1. Introduction

We live in a world brimming with uncertainty, where we constantly have to make a lot of decisions with incomplete information. How we make decisions is truly an enigma. Classical decision theory [1–5] is a “black box”; we do not know what really happens inside the box. The human behaviors exhibited during decision making, such as the order effect, cannot be sufficiently explained by decision theory based on classical probability. Scientists are trying to apply quantum theory to reveal how decisions are made. Recently, many quantum-like decision theories [6–8] have been proposed based on quantum probability to revise the mathematical structure that is used in classical models. Aerts et al. first proposed to apply quantum probability in decision theory [9,10]; Busemeyer et al. proposed a quantum-like model to describe human judgments and the order effect [11–13]; Khrennikov et al. improved the Busemeyer quantum model by applying quantum instruments of quantum measurement theory [14–18]; Yukalov et al. proposed a rigorously axiomatic quantum decision theory [19–21]; Xin et al. proposed a quantum value operator decision theory [22].

Whether it is classical decision theories or quantum-like decision theories, all well-developed decision models have applied a rigorous mathematical structure to describe people’s decision making under uncertainty. We are firm believers that people’s subjective belief cannot be computed by rigorous mathematical formula. The main issue with mathematical models is that they are difficult to understand, cannot reflect the dynamic changes in the state of the decision-makers’ mind, and it is not easy to calculate theoretical values to compare with actual observed outcomes when the mathematical model becomes more complex.

In this paper, based on Darwin’s natural selection, we propose an algorithm that incorporates insights from quantum theory to describe people’s decision making under uncertainty. Our decision model emphasizes machine learning, where decision-makers build-up their experience by being rewarded or punished for each decision they make,



Citation: Xin, L.; Xin, K.; Xin, H. On Laws of Thought—A Quantum-like Machine Learning Approach. *Entropy* **2023**, *25*, 1213. <https://doi.org/10.3390/e25081213>

Academic Editors: Andrei Khrennikov and Irina Basieva

Received: 1 June 2023

Revised: 3 August 2023

Accepted: 9 August 2023

Published: 15 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

preparing them to make better decisions in the future. This is more in line with decision-makers in the real world.

Our proposed quantum-like decision theory discovers laws of thought by machine learning an observed time series; there is no differential equation, and no transition probability computation in our decision theory. We do not model with the usual utility function or observables of the projection-type in other quantum-like decision theories, but a logic tree and a value tree. The logic tree determines the state of each point in the time series, and the value tree calculates the absolute value between two points in the time series. A logic tree and value tree work together to depict the entire decision-making process of a decision-maker.

In this paper, a “machine economist” is developed and the Dow Jones index is used as historical data for training the “machine economist”. The “machine economist” will trade Dow Jones index futures, build up experience, optimize a set of trading strategies to maximize profits, and finally construct a theory about financial markets.

2. Quantum-like Machine Learning Algorithm

The change in the Dow Jones index over time can be defined in terms of a time series consisting of states and observable values in (1).

$$\{(q_k, x_k)\}_{k=1, \dots, N} \quad (1a)$$

$$q_k = \begin{cases} 0, & x_k \geq x_{k-1} \\ 1, & x_k < x_{k-1} \end{cases} \quad (1b)$$

where q_k denotes the dynamic state of the Dow Jones index; if the closing price of the Dow Jones index goes up, then the state is 0 ($q_k = 0$); if the closing price of the Dow Jones index goes down, then the state is 1 ($q_k = 1$); x_k denotes the observed value (closing price) of the Dow Jones index; data sequences $\{x_k, k = 1, \dots, N\}$ describe the trajectory of the Dow Jones index.

Time series $\{(q_k, x_k)\}$ can be considered as questions posed by the market, in which “machine economist” need to describe and interpret the market based on observed data sequences. The “machine economist” is actually playing with the market, and the market is neither optimistic nor pessimistic, but is just playing dice with the “machine economist”. The “machine economist” tries to maximize the expected value in order to find the most probabilistically correct answers (maximize profits).

The question now becomes: Can the “machine economist” find an answer?

Here, a quantum-like machine learning algorithm is proposed to answer the questions posed by the market. A quantum-like machine learning algorithm can be expressed in terms of two-part trees: the first part is a logic tree, which applies “yes or no” answers to determine the dynamically changing state of the security, and the other part is a value tree that describes the closing prices of the security. Together, the logic tree and value tree will reconstruct the trajectory of the security.

1. Logic tree: to determine the action to be taken and calculate the theoretical value of the closing price.
2. Value tree: to calculate the absolute value of the difference in closing prices between two trading points of the Dow Jones index.

The goal of an algorithm A_k in general is to be able to either:

- (1) Generate the results to match the observed outcomes;
- (2) Predict the next outcome.

In other words, given a sequence of data $\{(q_k, x_k)\}$ as an input by the market, a “machine economist” develops an algorithm A_k to output a sequence of data $\{(q'_k, x'_k)\}$:

$$\{(q_k, x_k)\} \xrightarrow{\text{input}} A_k(\text{logicTree}, \text{valueTree}) \xrightarrow{\text{output}} \{(q'_k, x'_k)\} \quad (2a)$$

Meet the following formula:

$$q'_k = q_k \text{ and } x'_k = x_k, \quad k = 1, 2, \dots, n \quad (2b)$$

$$q'_{n+1} = q_{n+1} \text{ and } x'_{n+1} = x_{n+1} \quad (2c)$$

Genetic programming (GP) [23–26] is used by the “machine economist” to search for a satisfactory algorithm. Just as the genes of the fittest of each species are passed down from generation to generation through natural selection, evolution algorithms can perform the same action through machine learning.

The idea and steps of GP are simple:

- (1) Randomly generate 300 logic or value trees;
- (2) Historical data is learned to obtain the fitness of each tree;
- (3) The satisfactory logic or value tree is obtained through the Darwinian principle of survival of the fittest (crossover, mutation and selection) after about 80 generations of evolution.

The GP Algorithm (Algorithm 1) is as follows:

Algorithm 1. GP Algorithm

Input:

- Historical dataset $\{(q_k, x_k), k = 0, \dots, N\}$ (each sample consists of a security’s state and closing price);
- Setting:
 - (1) Operation set F;
 - (2) Dataset T;
 - (3) Crossover probability = 70%; Mutation probability = 5%.

Initialization:

- Population: randomly create 300 individuals.

Evolution:

Loop: for $i = 0$ to 80 generations:

- a Calculate fitness for each individual based on the historical dataset;
- b According to the quality of fitness:
 - i Selection: selecting parents.
 - ii Crossover: generate a new offspring using the roulette algorithm based on crossover probability.
 - iii Mutation: randomly modify the parent based on mutation probability.

Output:

- An individual of the best fitness.
-

2.1. Value Tree

A value tree is a traditional function tree. The final form of the value tree is represented as a function. The output from this function is a numeric value. For a value tree the operation set F and dataset T are as follows

- (1) Operation set $F = \{+, -, \times, \div, \log, \exp\}$;
- (2) Dataset $T = \{t, fl, av, h, l\}$.

where t denotes the t -th trade of the Dow Jones index time series; fl denotes the average fluctuation of the closing price; av denotes the average closing price; h denotes the highest closing price; l denotes the lowest closing price.

A value tree is a function that consists of operation set F and dataset T .

$$\text{valueTree} = f(F, T) \tag{3}$$

We define the absolute value of the Dow Jones index between two trading points as follows:

$$d_{t,t-1} = |x_t - x_{t-1}| \tag{4}$$

The “machine economist” can calculate the absolute value between two trading points using the value tree:

$$d'_{t,t-1} = f(F, \{t, fl, av, h, l\}) - f(F, \{t - 1, fl, av, h, l\}) \tag{5}$$

Now we can define the fitness function for the value tree as follows:

$$\text{fitness}_{\text{valueTree}} = -\sum_{k=1}^n (d'_{t,t-1} - d_{t,t-1})^2 \tag{6}$$

where $d'_{t,t-1}$ is the absolute value calculated by the value tree in (5), and $d_{t,t-1}$ is the observed absolute value of the market in (4). The fitness function is essentially a particular type of function that is used to summarize, as a single figure of merit, how close a given design solution is to achieving the set aims. Fitness functions are used in GP to guide simulations towards optimal design solutions. In order to reach the optimal solution, the GP algorithm implements a continuous evolution process through selection, crossover, and mutation. The goal of continuous evolution is to find a satisfactory value tree that makes $d'_{t,t-1}$ as close to $d_{t,t-1}$ as possible.

2.2. Logic Tree

A logic tree is a matrix tree constructed from eight basic quantum gates. The final form of a logic tree is represented as a matrix. The output from this matrix is a vector (an action, for example, buy or sell). The purpose of the logic tree is to simulate the decision-making process of the “machine economist”. Table 1 shows that the Dow Jones index has two states: q_1 (index up) and q_2 (index down); the “machine economist” has two possible actions to take, a_1 (buy) and a_2 (sell). $p_1|x, p_1|-x, p_2|x, p_2|-x$ are four possible outcomes determined by both the market and the “machine economist”. For example, $p_1|x$ means that the “machine economist” takes action a_1 (buy) with a subjective probability of p_1 and makes a profit x amount of money because the index is up (q_1); $p_2|-x$ indicates that the “machine economist” takes action a_2 (sell) with a subjective probability of p_2 and loses x amount of money because the index is up (q_1).

Table 1. State–action–value decision table.

Action	State	q_1	q_2
	a_1		$p_1 x$
a_2		$p_2 -x$	$p_2 x$

The market influences the traders’ decisions, while all traders’ actions then decide the market’s state. This interaction between the two, the objective (state of the market) and the subjective (traders’ beliefs), is what causes both the result of the decisions (gain or loss) and the state of the market (up or down) to be uncertain.

The state of the market describes the objective world; it can be represented by the superposition of all possible states in terms of the Hilbert state space as shown below [27,28].

$$|\psi\rangle = c_1 |q_1\rangle + c_2 |q_2\rangle \tag{7}$$

where $|q_1\rangle$ denotes a state in which the market has increased, and $|q_2\rangle$ denotes a state in which the market has decreased. $|c_1|^2$ is the objective frequency of the increase; $|c_2|^2$ is the objective frequency of the falling market.

The state of the trader’s mind is the subjective world. We postulate that when the trader is undecided in making a trade (buy or sell), it can be represented by superposition of all possible actions as follows.

$$|\phi\rangle = \mu_1 |a_1\rangle + \mu_2 |a_2\rangle \tag{8}$$

where $|a_1\rangle$ denotes the trader’s action to buy, and $|a_2\rangle$ denotes the trader’s action to sell. $p_1 = |\mu_1|^2$ is the trader’s degree of belief in betting that the market will rise; $p_2 = |\mu_2|^2$ is trader’s degree of belief in betting that the market will fall.

The information available to the “machine economist” prior to making its decision is incomplete; the “machine economist” does not know whether the market will rise or fall, forcing the “machine economist” to essentially guess. Before a “machine economist” makes a decision, its mind state is in a pure state, a superposed state in which it can decide whether to buy and sell at the same time. However, in reality, the “machine economist” cannot take an action to buy and sell simultaneously. This pure state is when the states of buy and sell are superposed in the “machine economist’s” mind. Then, when the “machine economist” makes the decision, the state of the “machine economist’s” mind then transforms from that pure state ρ into a mixed state ρ' , which is when it decides to buy or sell, with certain degrees of belief. Basically, this transformation is the “machine economist” choosing from one of the available actions, with action a_1 being buy with probability p_1 and action a_2 being sell with probability p_2 , shown below.

$$\text{Tradingprocess : } \rho = |\phi\rangle\langle\phi| \xrightarrow{\text{decision}} \rho' = p_1 |a_1\rangle\langle a_1| + p_2 |a_2\rangle\langle a_2| \tag{9}$$

It is expressed in matrix form as follows:

$$\rho = \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} \xrightarrow{\text{diagonalization}} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \xrightarrow{\text{normalization}} \rho' = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \tag{10a}$$

$$= p_1 |a_1\rangle\langle a_1| + p_2 |a_2\rangle\langle a_2|$$

$$|a_1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |a_2\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; |a_1\rangle\langle a_1| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, |a_2\rangle\langle a_2| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{10b}$$

The pure state (quantum density matrix) ρ can be approximately constructed from eight basic quantum gates. For a logic tree the operation set and dataset are as follows:

- (1) Operation set $F = \{+, *, //\}$;
- (2) Dataset $T = \{H, X, Y, Z, S, D, T, I\}$

$$\left\{ \begin{array}{l} H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad D = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \right\}$$

where + means two matrices are added, * means two matrices are multiplied, and // means that one is randomly selected from two branches. H, X, Y, Z, S, D, T, I are eight basic quantum gates (2×2 matrix) [29,30].

A logic tree is composed of operation set F and dataset T that determines the action taken by the “machine economist” and calculates the closing price of the Dow Jones index at the different trading points.

$$\text{logicTree} = f(F, T) \tag{11}$$

With a logic tree the “machine economist” can decide the action to be taken a_t and the closing price x'_t ($d'_{t,t-1}$ can be calculated by the value tree in (5)):

$$a_t = \text{logicTree}(F, T) = \begin{cases} 0, & \text{buy is excuted (degrees of belief is } p_1) \\ 1, & \text{sell is excuted (degrees of belief is } p_2) \end{cases} \quad (12a)$$

$$x'_t = \begin{cases} x'_{t-1} + d'_{t,t-1}, & \text{if } a_t = 0 \\ x'_{t-1} - d'_{t,t-1}, & \text{if } a_t = 1 \end{cases} \quad (12b)$$

The next step is to find a way to optimize the logic tree with a group of satisfactory strategies to guide the “machine economist’s” decisions. To optimize anything, there needs to be: first, a selection of a good evaluation function, and two, how to acquire an optimal solution. First off, in our model, the “machine economist” will try to maximize its expected value when making any trading decisions. Thus, we need to evaluate how “fit” the result (profit or deficit) of the “machine economist’s” decision are, which can be performed using the expected value in (13) as a fitness function to optimize the logic tree by evolving them. The whole idea of having GP go through an iterative evolution loop is to find a satisfactory logic tree by means of learning historical data to obtain the most optimal solution. The learning rules are as follows:

- (1) If the Dow Jones index is up (q_1):
 - i. If the “machine economist” bets the Dow Jones index is up to buy ($a_1 = 0$), it profits ($d'_{t,t-1}$);
 - ii. If the “machine economist” bets the Dow Jones index is down to sell ($a_2 = 1$), it deficits ($-d'_{t,t-1}$).
- (2) If the Dow Jones Index is down (q_2):
 - i. If the “machine economist” bets the Dow Jones index is down to sell ($a_2 = 1$), it profits ($d'_{t,t-1}$);
 - ii. If the “machine economist” bets the Dow Jones index is up to buy ($a_1 = 0$), it deficits ($-d'_{t,t-1}$).

The expected t-th value of the “machine economist” is as follows:

$$EV_t = \begin{cases} = p_1 d'_{t,t-1}, & \text{market is up and the “machine economist” buys with degrees of belief } p_1 \\ = -p_2 d'_{t,t-1}, & \text{market is up and the “machine economist” sells with degrees of belief } p_2 \\ = -p_1 d'_{t,t-1}, & \text{market is down and the “machine economist” buys with degrees of belief } p_1 \\ = p_2 d'_{t,t-1}, & \text{market is down and the “machine economist” sells with degrees of belief } p_2 \end{cases} \quad (13)$$

Now we can define the fitness function for the logic tree as follows:

$$\text{fitness}_{\text{logicTree}} = \sum_{t=1}^n EV_t \quad (14)$$

$\text{fitness}_{\text{logicTree}}$ maximizes “machine economist” expectations ($\max_{\text{logicTree}} (\sum_{t=1}^n EV_t)$); $\text{fitness}_{\text{valueTree}}$ applies negative feedback to make ($d'_{t,t-1} \xrightarrow{\text{equal}} d_{t,t-1}$). Logic tree together with value tree will reconstruct the trajectory of the Dow Jones index $\{(q'_k, x'_k)\} \xrightarrow{\text{equal}} \{(q_k, x_k)\}$, and make a prediction about the future outcomes as follows:

$$d'_{n+1,n} = \text{valueTree}(F, \{n + 1, fl, av, h, l\}) - \text{valueTree}(F, \{n, fl, av, h, l\}) \quad (15a)$$

$$a_n = \text{logicTree}(\{+, \times, //\}, \{H, X, Y, Z, S, D, T, I\}) = \begin{cases} 0, & \text{“machine economist” takes action } a_1 \text{ (buy)} \\ 1, & \text{“machine economist” takes action } a_2 \text{ (sell)} \end{cases} \quad (15b)$$

$$x'_{n+1} = \begin{cases} x'_n + d'_{n+1,n}, & \text{if } a_n = 0 \\ x'_n - d'_{n+1,n}, & \text{if } a_n = 1 \end{cases} \quad (15c)$$

3. Results

The Dow Jones index from 1 to 30 December 2022 is used for training the “machine economist” as shown in Table 2. The first column indicates the state, where 0 means the index is up and 1 means the index is down; the second column indicates the closing price of the index. The first row is the initial condition, −1 indicates the state is uncertain, and 34,395.01 indicates the base closing price for machine learning.

Table 2. The Dow Jones index (1–30 December 2022).

−1	34,395.01
0	34,429.88
1	33,947.1
1	33,596.34
0	33,597.92
0	33,781.48
1	33,476.46
0	34,005.04
0	34,108.64
1	33,966.35
1	33,202.22
1	32,920.46
1	32,757.54
0	32,849.74
0	33,376.48
1	33,027.49
0	33,203.93
0	33,241.56
1	32,875.71
0	33,220.8
1	33,147.25

3.1. Dow Jones Index’s Value Tree

By applying the fitness function of $\text{fitness}_{\text{valueTree}}$ (6), the “machine economist” can continuously learn the historical data to evolve a satisfactory value tree, as shown in Figure 1.

$$\text{valueTree}_{\text{dowJones}} = (t * \text{fl}) \quad (16)$$

where t denotes the t -th trade of the Dow Jones index, and fl denotes the average fluctuation of the closing price. The “machine economist” can use this value tree ($\text{valueTree}_{\text{dowJones}}$) to calculate the absolute value between two trading points of the Dow Jones index:

$$d'_{t,t-1} = t * \text{fl} - (t - 1) * \text{fl} = \text{fl} = 265 \quad (17)$$

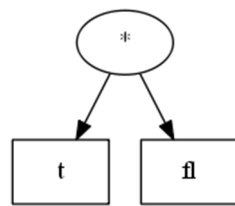


Figure 1. Value tree for the Dow Jones index.

3.2. Dow Jones Index’s Logic Tree

By applying the fitness function $fitness_{logicTree}$ (14) and $d'_{t,t-1} = 265(17)$, the “machine economist” can continuously learn the historical data of the Dow Jones index, and evolve a satisfactory logic tree as shown in Figure 2.

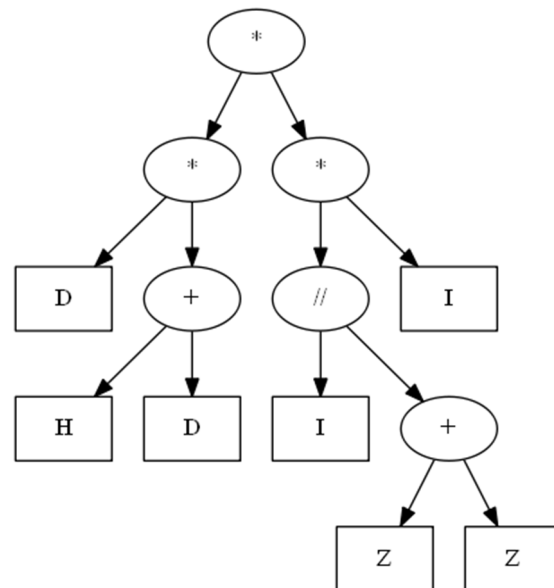


Figure 2. Logic tree for the Dow Jones index.

Figure 3 shows the optimization curve of the evolution algorithm. Figure 4 shows the learning curve of the evolution algorithm. As shown in Figure 4, the expected value of this logic tree is 5101, very close to the maximum expected value of the Dow Jones index time series (1–30 December 2022), which is 5308.

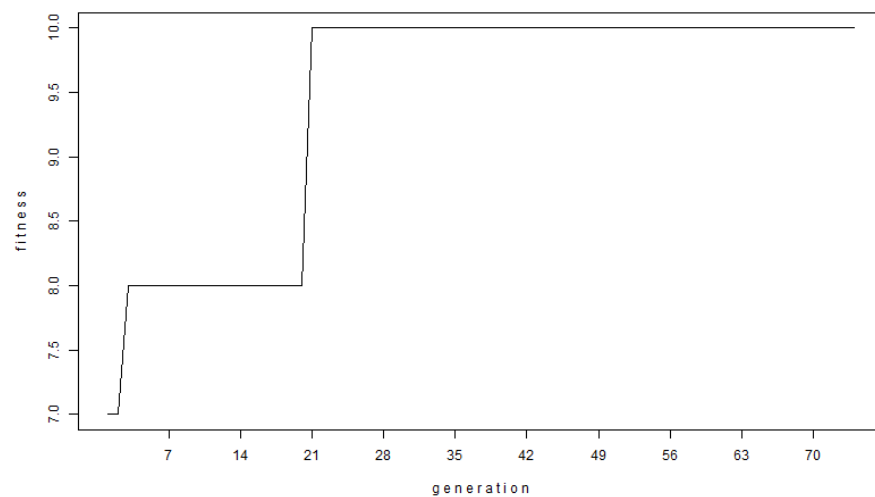


Figure 3. The optimization curve of the evolution algorithm.

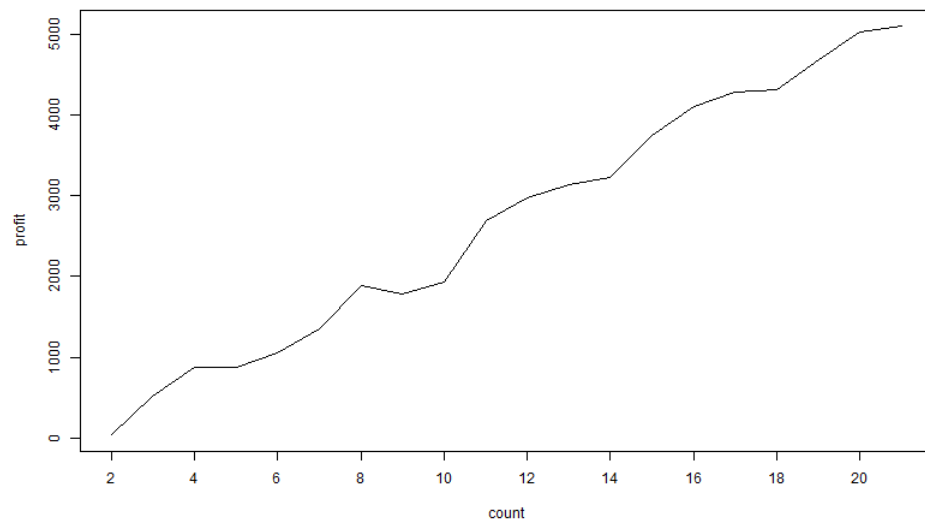


Figure 4. The learning curve of the evolution algorithm.

The $\text{logicTree}_{\text{dowjones}}$ (18) provides two strategies $\{S_1, S_2\}$, and the “machine economist” can randomly choose a strategy from the two and apply the strategy chosen to guide the “machine economist” in choosing which action to take (buy or sell with a subjective degrees of belief). If strategy S_1 is chosen, then the “machine economist” is 100% sure that the index is up (buy); if strategy S_2 is chosen, the “machine economist” is 100% sure that the index is down (sell). Combining $\text{logicTree}_{\text{dowjones}}$ (18) and $d'_{t,t-1}$ (17), the “machine economist” can determine the action to be taken a_t (19) and calculate the closing price x'_t (20) of the Dow Jones index.

$$\text{logicTree}_{\text{dowjones}} = ((D * (H + D)) * ((I / (Z + Z)) * I)) \tag{18}$$

- $S_1 = ((D * (H + D)) * ((Z + Z) * I)) \rightarrow |a_1\rangle\langle a_1| (p_1 = 100\%, p_2 = 0)$
- $S_2 = ((D * (H + D)) * (I * I)) \rightarrow |a_2\rangle\langle a_2| (p_1 = 0, p_2 = 100\%)$

$$a_t = \begin{cases} 0, & \text{if } S_1 \text{ is selected} \\ 1, & \text{if } S_2 \text{ is selected} \end{cases} \tag{19}$$

$$x'_t = \begin{cases} x'_{t-1} + 265, & \text{if } a_t = 0 \text{ (buy)} \\ x'_{t-1} - 265, & \text{if } a_t = 1 \text{ (sell)} \end{cases} \tag{20}$$

By applying Equations (19) and (20), the “machine economist” can then reconstruct the trajectory of the Dow Jones index, as shown in Figure 5 with 19 wins and 1 loss. The $\text{logicTree}_{\text{dowjones}}$ randomly selects strategy S_1 and strategy S_2 with 9 buy events and 11 sell events, as shown in Figure 6, very close to the market 10 times up and 10 times down. In Figure 6, a positive bar represents a buy action taken by the logic tree, and 100 means with 100% degrees of belief to buy; a negative bar represents a sell action taken by the logic tree, and -100 means with 100% degrees of belief to sell. The random actions taken by $\text{logicTree}_{\text{dowjones}}$ to buy and sell imply that the evolution algorithm believes the market is efficient, i.e., the market is walking randomly.

Although the “machine economist” approximately reconstructs the price trajectory of the index, it can only make a 50/50 probability prediction of the future state of the Dow Jones index (up or down) by randomly choosing strategy 1 (believe the index is up) or strategy 2 (believe the index is down), i.e., without using partial differential equations and joint probabilities. The “machine economist” independently discovers that the market is efficient (random walk).

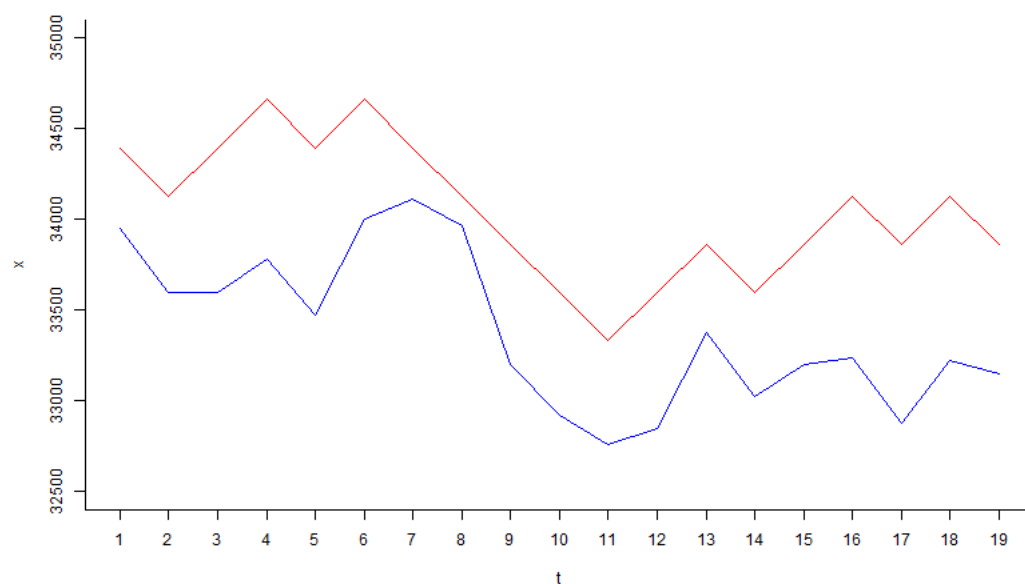


Figure 5. Trajectory of the Dow Jones index (blue line is the observed closing price, and the red line is the computed closing price).

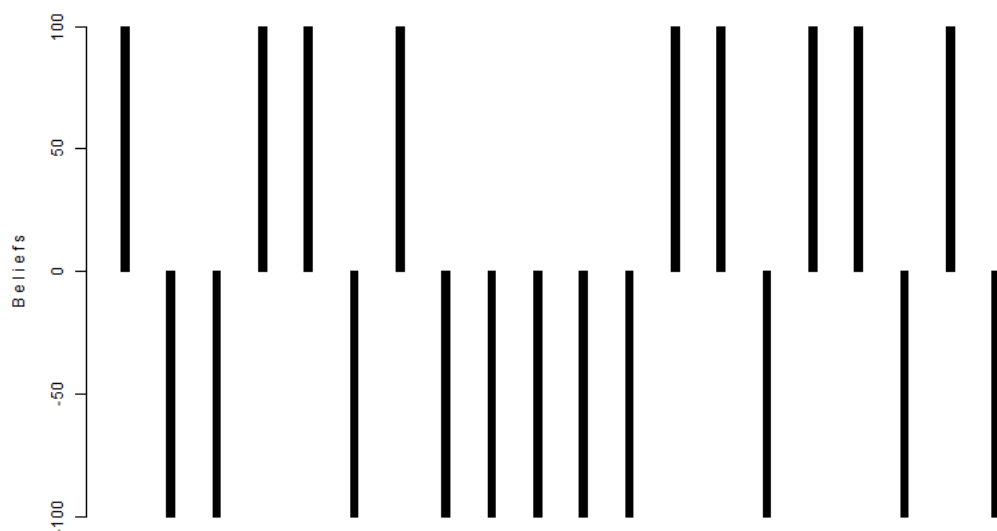


Figure 6. The buy (positive) or sell (negative) actions taken by the logic tree with degrees of belief (buy: 100; sell: -100).

4. Discussion

More than a hundred years ago, Louis Bachelier found the similarity between stock price movement and Brownian motion by studying the Paris stock market data, and Bachelier applied a normal distribution to describe the movement in stock prices using stochastic differential equations. In this paper, the “machine economist” constructs a theory about the financial markets by studying a Dow Jones index time series, where the “machine economist” applies an algorithm and treats the data structure of the market as unknown, and the “machine economist” discovers that the market is efficient (random walk) through machine learning. It should be emphasized here that, unlike Bachelier, the “machine economist” discovers the efficient market hypothesis through machine learning with the evolution algorithm, without using any stochastic differential equations or rational economic man hypotheses.

Based on the superposition principle of quantum mechanics, we introduce objective (market) and subjective (trader) dual uncertainty to decision theory (Equations (7) and (8)). “Quantum jump” is applied to explain the decision process: that is, the decision process is

a projection from a pure state to a mixed state (Equation (9)). A quantum density matrix in a pure state ($\rho = |\phi\rangle\langle\phi|$) has quantum interference, such as Schrödinger's cat who is dead and alive at the same time, a market that is up and down at the same time or a trader who can buy and sell at the same time; the mixed state is the classic statistical state, that is, the market can only be up or down or a trader can only buy or sell. Furthermore, we used eight fundamental quantum gates to construct a quantum density matrix (pure state), and optimized the quantum density matrix through evolutionary algorithms.

Time series in the complex real world rarely have a certain probability distribution, so the key is not to find the probability distribution from the time series but to find valuable information (experience or knowledge) and apply the learned experience to make decisions. In this paper, the "machine economist" uses both a logic tree and a value tree together to study historical data in order to obtain useful information (information of the state and the absolute value between two trading points). Instead of fitting a curve (price fluctuations of the Dow Jones index) with just one equation, the "machine economist" first uses the value tree to find the absolute value of the price difference between two trading points, and then uses the logic tree to determine the action to be taken (with degrees of belief). The value tree (objective) and logic tree (subjective) fit the curve together.

Author Contributions: All authors conducted the research and contributed to the development of the model. H.X. contributed as an expert in quantum theory and non-linear science. L.X. contributed to the research from the aspects of machine learning, decision theory and wrote the code. L.X. and K.X. wrote this manuscript and performed the data analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request.

Acknowledgments: We would like to thank the anonymous referees of this journal, whose comments substantially improved the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Von Neumann, J.; Morgenstern, O. *Theory of Games and Economic Behavior*; Princeton University Press: Princeton, NJ, USA, 1944.
2. Savage, L.J. *The Foundation of Statistics*; Dover Publication Inc.: New York, NY, USA, 1954.
3. Binmore, K. *Rational Decisions*; Princeton University Press: Princeton, NJ, USA, 2009.
4. Kahneman, D.; Tversky, A. Prospect theory: An analysis of decision under risk. *Econometrica* **1979**, *47*, 263–292. [[CrossRef](#)]
5. Simon, H.A. *Reason in Human Affairs*; Stanford University Press: Stanford, CA, USA, 1983.
6. Ashtiani, M.; Azgomi, M.A. A survey of quantum-like approaches to decision making and cognition. *Math. Soc. Sci.* **2015**, *75*, 49–80. [[CrossRef](#)]
7. Busemeyer, J.R.; Bruza, P.D. *Quantum Models of Cognition and Decision*; Cambridge University Press: Cambridge, UK, 2012.
8. Haven, E.; Khrennikov, A. *Quantum Social Science*; Cambridge University Press: Cambridge, UK, 2013.
9. Aerts, D.; Aerts, S. Applications of quantum statistics in psychological studies of decision processes. *Found. Sci.* **1995**, *1*, 85–97. [[CrossRef](#)]
10. Aerts, D. Quantum structure in cognition. *J. Math. Psychol.* **2009**, *53*, 314–348. [[CrossRef](#)]
11. Busemeyer, J.; Franco, R. What is the evidence for quantum like interference effects in human judgments and decision behavior? *NeuroQuantology* **2010**, *8*, S48–S62. [[CrossRef](#)]
12. Busemeyer, J.R.; Franco, R.; Pothos, E.M. Quantum probability explanations for probability judgment errors. *Psychol. Rev.* **2010**, *118*, 193. [[CrossRef](#)] [[PubMed](#)]
13. Wang, Z.; Busemeyer, J.R. A quantum question order model supported by empirical tests of an a priori and precise prediction. *Top. Cogn. Sci.* **2013**, *5*, 689–710.
14. Khrennikov, A.; Basieva, I.; Dzhamfarov, E.N.; Busemeyer, J.R. Quantum models for psychological measurements: An unsolved problem. *PLoS ONE* **2014**, *9*, e110909. [[CrossRef](#)] [[PubMed](#)]
15. Asano, M.; Basieva, I.; Khrennikov, A.; Ohya, M.; Tanaka, Y. A quantum-like model of selection behavior. *J. Math. Psychol.* **2017**, *78*, 2–12. [[CrossRef](#)]
16. Basieva, I.; Khrennikova, P.; Pothos, E.M.; Asano, M.; Khrennikov, A. Quantum-like model of subjective expected utility. *J. Math. Econ.* **2018**, *78*, 150–162. [[CrossRef](#)]

17. Ozawa, M.; Khrennikov, A. Application of theory of quantum instruments to psychology: Combination of question order effect with response replicability effect. *Entropy* **2019**, *22*, 37. [[CrossRef](#)] [[PubMed](#)]
18. Ozawa, M.; Khrennikov, A. Modeling combination of question order effect, response replicability effect, and QQ-equality with quantum instruments. *J. Math. Psychol.* **2021**, *100*, 102491. [[CrossRef](#)]
19. Yukalov, V.I.; Sornette, D. Physics of risk and uncertainty in quantum decision making. *Eur. Phys. J. B* **2009**, *71*, 533–548. [[CrossRef](#)]
20. Yukalov, V.I.; Sornette, D. Quantum probabilities as behavioral probabilities. *Entropy* **2017**, *19*, 112. [[CrossRef](#)]
21. Yukalov, V.I. Evolutionary Processes in Quantum Decision Theory. *Entropy* **2020**, *22*, 681. [[CrossRef](#)] [[PubMed](#)]
22. Xin, L.; Xin, H. Decision-making under uncertainty—A quantum value operator approach. *Int. J. Theor. Phys.* **2023**, *62*, 48. [[CrossRef](#)]
23. Holland, J. *Adaptation in Natural and Artificial System*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
24. Goldberg, D.E. *Genetic Algorithms—In Search, Optimization and Machine Learning*; Addison-Wesley Publishing Company, Inc.: New York, NY, USA, 1989.
25. Koza, J.R. *Genetic Programming, on the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992.
26. Koza, J.R. *Genetic Programming II, Automatic Discovery of Reusable Programs*; MIT Press: Cambridge, MA, USA, 1994.
27. Von Neumann, J. *Mathematical Foundations of Quantum Theory*; Princeton University Press: Princeton, NJ, USA, 1932.
28. Dirac, P.A.M. *The Principles of Quantum Mechanics*; Oxford University Press: Oxford, UK, 1958.
29. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2000.
30. Benenti, G.; Casati, G.; Strini, G. *Principles of Quantum Computation and Information I*; World Scientific Publishing: Singapore, 2004.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.