International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064

A Distributed Intelligent Framework for Handling Dynamically Arriving Information

Shanthi. P[1], Saraswathy. K[2]

[1] Assistant Professor, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore, TamilNadu, India

[2] M.Phil Research Scholar, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore, TamilNadu, India

Abstract: ***When there is an outlay to waiting for more information, the question is when to make the proper decision. This paper deals with the stock values in trading as dynamic information. We propose a system which gives suggestions or tips for deciding when to sell or buy the stocks and whether to stop the process or wait when the market rate changes. The influence of dynamic information may depend on waiting cost and stop function. The existing system uses a model that characterizes the influence of dynamic information on the utility of the decision. With the help of an agent the share holder can get the status of his/her stock and based on that the decision can be taken. An optimal algorithm has been used to guarantee the best time to stop, but its complexity is exponential in the number of candidates. So an alternative framework in which the different candidates are solved separately to reduce the complexity. The alternative framework leads to a range of specific heuristic algorithms.***

Keywords: Agent; Multiagent; decision making; dynamic information;

1. Introduction

An agent is a computing entity that performs user delegated tasks autonomously. An agent is the one that is authorized to act for another. Agents possess the characteristics of delegacy, competency, and amenability**.** Examples of human agents include booking agents, Sales agents, and Politicians.A Software Agent is an artificial agent which operates in a software environment. Software environments include operating systems, computer applications, databases, networks, virtual domains. Delegacy for software agents centers on persistence. Software agents stay resident, or persistent, as background processes after being launched. By making decisions and acting on their environment independently, software agents reduce human workload by generally only interacting with their end-clients when it is time to deliver results. Additionally, autonomous automation can lead to super-human performance in terms of volume and speed. Competency within a software environment requires knowledge of the specific communication protocols of the domain. These protocols must be preprogrammed into the software agents, limiting their useful range. Amenability for non-intelligent software agents is generally limited to providing control options and the generation of status reports that require human review. Such agents often tend to be brittle in the face of a changing environment, necessitating a modification of their programming to restore performance.

Intelligent Software Agent (ISA) is a software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients. Artificial Intelligence is the imitation of human intelligence by mechanical means. Clients can reduce the human workload by delegating the tasks to ISAs that normally would require human-like intelligence. Thus the word "agent" by itself generally means ISAs in the terms of the present-day research community. An intelligent agent is a computer system that is capable of flexible autonomous action in order to meet its design objectives. The word flexible means that the system must be responsive, proactive and social. Agents should perceive their environment which may be the physical world, a user, a collection of agents, the Internet, etc. and respond in a timely fashion to the changes that occur in it. Agents should not simply act in response to their environment; they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate. The agents should be able to interact, when they are found appropriate to work with other artificial agents and humans in order to complete their own problem solving and to help others with their activities. In addition to communicating with their environment to collect data and actuate changes, ISAs can often analyze the information to find non-obvious or hidden patterns, extracting knowledge from raw data. Environmental modes of interaction are richer, incorporating the media of humans such as natural language text, speech, and vision. Amenability in ISAs can include self-monitoring of achievement towards the client goals combined with continuous, online learning to improve performance. Adaptive mechanism in ISAs means that they are far less brittle to changes in the environment and may actually improve. In addition, client responsiveness may go so far as to infer what a client wants when the client himself does not know or cannot adequately express the desired goals in definitive terms. This paper deals with the agents which are capable to handle dynamically arriving information. The dynamic information considered in this paper is the stock values in trading**.**

2. Problem Statement

There are many applications for a simple agent. Simple agents have been designed for different applications which share common features. Individual agents are designed and built to enact particular roles. These agents are autonomous, goal directed entities, which are responsive to their environment. They must typically interact with other agents in order to carry out their role. Such interactions are a natural consequence of certain interdependencies which

exist between the agents, their environment, and their design objectives. As the agents are autonomous, the interactions are usually fairly sophisticated involving cooperation, coordination and negotiation. Two important observations have to be made about developing agent based applications. The first observed fact is that the detailed problem solving actions of the agent can only be determined at runtime. Individual behavior is regulated by a complex interplay between the agent's internal state and its external influences or its environment and the other agents. Since the behavior of individual agents is not uniquely determined at design time, the behavior of the system as a whole can also only emerge at run time. These points have some obvious implications for the use of an agent-based approach in safety-critical application domain such as air traffic control, where it is essential that the system satisfies its specification. There are different dimensions that can be used for analyzing the agent-based systems. Single-agent systems are in a sense much simpler than multi-agent systems, since they do not require the designer to deal with issues such as cooperation, negotiation, and so on. Simple agents can not give best results when the number of candidates is high and the response time also gets increased. When the number of users increased then some time delay could be occurred to make a decision whether to stop or continue the trading.

3. Related Works

A multi agent based hybrid system that is using multi criteria decision-making and automated negotiation. In that, Business negotiation is a decision-making process that seeks to find an agreement, which will satisfy the requirements of two or more parties in presence of limited information, and conflicting preferences. In that hybrid systems that are able to have an appropriate performance in real conditions of different types of automated negotiations.

An agent-based adaptive Architecture had been proposed to extend Moodle (Modular Object-Oriented Dynamic Learning Environment) in order to support instructional decisions and adaptive behavior in engineering education. This paper described the characteristics, functions, and interactions of the agents which take part in each module of the adaptive architecture. In addition, they described the origin and function of ToDei, the proposed intelligent agent for Instructional Decisions Making. This agent is in charge of collecting information generated by the rest of agents and deciding what is the best for the final users, tutors and students, taking into account their attitudes towards the learning environment.

A multi-agent robot system, each robot must behave by itself according to its states and environments. In this method they using neural networks and particle swarm optimization (PSO) for the decision-making in the multi-agent robot system. In this system, a neural network is used for behavior decision controller. The inputs of the neural network are decided by the last actions of other robots. Then the outputs determine the next action that the robot will choose. The weight values imply the adaptiveness of robots in multi-agent robot system.

Another Multi-agent based fault diagnosis reference model for MSW incineration process is important for high speed and automation The desired high levels of incineration efficiency as well as plant facility reliability. Fault diagnosis and maintenance are vital aspects in MSW incineration process, in this sense, diagnosis and maintenance systems should support decision-making tools, new maintenance approaches and techniques, the enterprise thinking and flexibility. In Multi-agents based fault diagnosis reference model for MSW incineration process is presented which combines the existing models and multi-agents. This model is based on a generic framework using multi-agent systems for MSW on-line monitoring system; in this sense, the Fault diagnosis problem is viewed like a feedback control process and the actions are related to the decision-making in the scheduling of the preventive maintenance task and the running of preventive and corrective specific maintenance tasks.

4. Methodology

We propose a model for representing the arrival of dynamic information and its influence on the utilities of candidates. A heuristic framework has been developed for solving the problem in which different candidates are solved separately. This method formally analyzes the framework and show that the calculations it performs are exactly correct, and it approximates the true solution by considering different policies from the optimal algorithm. It translates to a suit of heuristic algorithms depending on how a key probability is computed. Complexity of a simple heuristic algorithm is only polynomial in the number of candidates. Compare the heuristic algorithm to the optimal algorithm and two more baseline algorithms. The One makes the decision in the beginning and the other that makes the decision after obtaining the whole information. Evaluate the algorithm in terms of quality of the utility of the decision and runtime.

4.1 Construction of Multi-Agent Application

[Stock Market Application] The proposed model creates an application for Stock market which includes multiple companies' daily share trading. This system is used to find out the causes that affect the rise and fall of stock prices in particular companies. In this module Agents can register themselves for providing information to the clients for buying and selling their products. Agents are programmed by the same team and they collaborate to complete task.

4.2 Modelling the Application with Candidate Utility Variables

Client can buy stock in the form of shares, which present a proportion of a company's assets and earnings. Client can buy or sell shares on the stock market. The user can buy and sell shares by going directly to the online share dealing. The stock market allows client to raise money by selling stock in a business to public and institutional investors.

4.3 Designing the Candidate Tree for the Candidate Utility Variables

The stock holder may want to identify the level of hands-on control over his/her stock trades. Some investors prefer to leave all trading decisions up to another individual, and others want to directly make transactions themselves. Agent will make advice to place an order for the stock he/she

wishes to purchase if agent wants to set the maximum price you are willing to pay for the shares. This ensures that the volatile market conditions do not fill the order at an unattractive price. This is a good way to create an order that may not be filled until a later date. It ensures client will get in at the price you want, but does not guarantee that the order will actually be filled should prices not trade at that level. Agent will suggest placing an order if he wants your account to execute a stock purchase only if the current price rises past a certain point. In this order, client specifies the share price that triggers the order. Client will use buy stop orders to automatically enter where a stock makes a substantial move in a new direction, surpassing prior highs.

4.4 Estimating the Events Gain Through the Optimal Algorithm

Some people, mostly beginners, trade their securities rather too often. It might be due to the tips they hear on TV or the advices by the friends. These things can make the user to feel that he/she needs to sell the current holdings and buy what's recommended. However, this doesn't work out. This will only manage to make the broker rich in the process because he rakes in trading fees. The optimal gain algorithm can be calculated by a decision tree approach. The optimal decision tree merges the candidate trees into a single decision tree whose depth is the maximal time of the variables in the candidate trees.

4.5 Developing a Framework Using Heuristic Algorithm for Multitasking the Application

Usually heuristic algorithms are used for problems that cannot be easily solved. To compute the probability exactly, the complexity of the algorithm might still be exponential in the number of candidates. Therefore we need to approximately compute this probability. A key step in the candidate-based algorithm is computing the probability that a candidate wins, given that a certain node in the candidate tree is reached.

4.6 Development of Decision Making Framework Using Naive Bayes Algorithm

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Among all kinds of investment activities, security's transaction is an important activity among all investors' involvements in the past decade. How to find out the relationships between a security's name, price, trading quantity, and/or other scientific technical indices, humane feeling, and how these factors affect the buy or the sell timing is an important condition to be a successful investor. An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. It is very easy to construct, not needing any complicated iterative parameter estimation schemes.

4.7 Performance Comparison

A Bayesian naive classifier is used to decide the future trends of a stock. Live Data records were collected, a total of 4 attributes were used in this classification process. Proposed Naive Bayes model shows a high estimated possibility of 40.54% without losing the investor's money by investing wrong targets. This result is helpful for those who have great interests to make profit in a stock market of similar situations.

5. Experimental Results

Based on the solution, it is possible to compare the results of the existing system and the proposed system. This framework compares a huge number of previous market rates of a particular stock in order to analyze well about the stock status. This helps the agent to give more accurate tips to the user to make a decision. The table Table.1 gives the performance comparison of the decisions that are taken by a single agent and with multiple agents for different companies that are having their stocks in the stock market.

Table 1: Trading comparison table

Company Id	*Trading Tips*	*Agent Predictions*	*NavieBayes Predictions*
1	Heuristic	58	75
2	Sell	0	61
3	Wait	58	66
4	Stop	46	0

The training data comparison given in the above table can be presented in a chart format. The Fig-1 depicts the comparison.

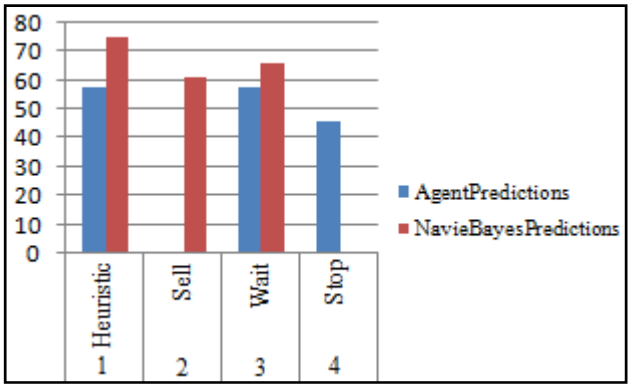

Figure 1: Number of candidate's vs. decisions

6. Conclusions

A classification paradigm is a data mining framework containing all the concepts extracted from the training dataset to differentiate one class from other classes existed in data. The primary goal of the classification frameworks is to provide a better result in terms of accuracy. However, in most of the cases we cannot get better accuracy particularly for huge dataset and dataset with several groups of data. When a classification framework considers whole dataset for training then the algorithm may become unusable because dataset consists of several group of data. The alternative way of making classification useable is to identify a similar

group of data from the whole training data set and then training each group of similar data. In our future work, we first split the training data using k-means clustering and then train each group with Naive Bayes Classification algorithm. In addition, we saved each model to classify sample or unknown or test data. For unknown data, we classify with the best match group/model and attain higher accuracy rate than the conventional Naive Bayes classifier.

References

[1] Ahmadi.K.D., Charkari.N.M., Multi agent based interactive recommendation and automated negotiation system in E-commerce November 2010.
[2] D.Jara-Roa, P. Valdiviezo-Díaz, M. Agila-Palacios, C. Sarango-Lapo, M. Rodriguez-Artacho., Conference Proceeding, An adaptive multi-agent based architecture for engineering education
[3] Peng, Liang , Hai Yun Liu, Decision-making and simulation in multi-agent robot system based on PSO-neural network. In IEEE International Conference on 2007.
[4] Hairui Wang , Kunming and Liu Yuping., Multi-agents Based Fault Diagnosis Systems in MSW Incineration Process. In ICMTMA,IEEE International Conference on 2010.
[5] Mirian, M.S., Ahmadabadi, M.N., Navabi, Z., A decision-making based approach for fault-handling in multi-agent systems in Proceedings of 9th international conference on 2002.
[6] Cossentino, M. , Lodato, C., Lopes, S. and Ribino, P., Multi agent simulation for decision making in warehouse management. In 2011 Federated Conference
[7] Eugene F. Fama and Marshall E.Blume, Filter Rules and Stock Market Trading. In Chicago Journals, Journal of Business.
[8] David L.Huff, Defining and Estimating Trading Area. In Journal of Marketing.
[9] Eugene F.Fama, Behaviour of Stock Market Prices, In Chicago Journals.
[10] Pattei Maes., Agents that Reduce Work and Information Overload, Camebridge University.