2020 International Conference on Computational Performance Evaluation (ComPE), North-Eastern Hill University, Shillong, Meghalaya, India. July 2–4, 2020

Decision Based Paradigm For Business Intelligence

Tejus R S
Computer Science Engineering
JSS Science and Technological University (Formerly SJCE)
Mysore, India
tejusrso@gmail.com

***Abstract*—The complexity of the business world has grown to large extents in the past decade. Business decisions and strategies have become more difficult to predict whether they achieve their long term goals or not. However, the business world represents a dynamic system that can be described as chaotic. This leads us to inculcating chaos theory in business strategy. This comparison is not straightforward and it needs a definitive system that is capable of representing concepts of chaos systems and delivers on the business side. In order to bridge these two parallels we use the help of a non-deterministic pushdown automaton. In this light, the present article aims to build a system which is not only capable of representing the business world, but also gives insights on the kind of decisions that need to be taken to ensure sustainability of the company.**

***Index Terms*—Business Intelligence, Chaos Theory, Pushdown Automata, Non-Deterministic Pushdown Automata**

I. Introduction

According to a study by Gartner, about 70-80 percent of business intelligence (BI) projects end up as a failure. In another study Gartner also mentions that 87 percent of organisations have low business intelligence maturity. These are unusually large number that is a major cause for concern. Business Intelligence is not a newborn concept, however it has lingered around for more than 35 years and despite the developments in technology, the success rate is abysmal.

This prompts us to ask questions such as should organisations invest in BI given its low success rate? The answer to this question is yes because every organisation relies on business intelligence to stay afloat in the competitive economic world. If that is the case then the next question that comes to mind is are we using the right approach and methods to obtain business intelligence? How can we tweak our approach and methods to obtain better results? This paper attempts at answering these questions as best as possible.

In this paper instead of using the consumer data and data from other conventional sources and performing statistical analysis over it, I have used solely history of decisions taken by organisations and their implications to build a model that can extract valuable information needed for business intelligence. In order to comprehend this, there is requirement to understand three important concepts namely Business Intelligence, Chaos theory, and Pushdown automata.

A. Business Intelligence

Business intelligence (BI) has been defined as the acquisition, interpretation, collation, assessment, and exploitation of business-related information (Chung, Chen, Nunamaker, 2003a).[1] It transforms available data into actionable insights which has the ability to drive business decisions. It includes different analytical tools which are used to understand data in a robust manner. Business intelligence is mainly used by enterprises to make informed decisions and strategise better. Usually, data collected from various sources (mostly customers) are analysed and informative patterns are observed. This information is used to make better decisions. The larger the data source available for analysis the more potential it has to provide valuable insights. However the problem with this approach is it becomes a search for a needle in a haystack, and more often than not you either don't really know what you are looking for or you don't find what you need. This jeopardises the credibility of the decisions that need to be taken to steer the company in the right direction. Suppose the data is unstructured or semi-structured, then this further reduces the possibility of a business intelligence project being successful as is the difficulty with dealing with unformatted data. [3]

B. Chaos theory

Chaos theory is a mathematical study of systems that are dynamic and exhibit disorder. However most of these systems have some deterministic laws governing them. These laws are not apparent and it can only be understood by thorough study of chaotic systems. Chaotic systems are systems that exhibit chaos such as fluid flow, weather and climate, stock market, economic behaviours, etc. All these systems have one important common feature which forms the corner stone of chaos theory. This common feature is the high sensitivity of all chaotic systems to its initial conditions. This sensitivity concludes that a minute change in the initial conditions will lead to large differences in future states. A small oversee such as an approximation in the initial state can yield largely different results from the required one in future states. Generally in chaotic systems the mathematical function that describes the system is performed iteratively over itself to obtain the future states. Hence the error accumulates and becomes exceedingly large as we move further. This property of chaotic systems makes it very difficult to obtain exact predictions. In order to arrive at the exact prediction of such system one needs to use highly sensitive and accurate instruments and methods to capture the initial conditions and reuse it at every stage to minimise the error. This is virtually impossible. However, this does not mean that this study is fruitless; instead this study brings us as close as possible to the truth of chaotic systems and hence makes the predictions as accurate as possible.[4]

C. Pushdown Automata

A pushdown automaton (PDA) is a type of automaton that uses a stack to help determine which state it transforms to next. The stack allows this automaton to have some memory. This ensures that a feature of the previous state can be used to choose the next state. The stack has two important operations pop and push. Depending on the encountered variable and states we can push a new symbol onto the stack or pop the topmost symbol from the stack. This allows the automaton to perform various transitions.

We use standard formal language notation:$\Gamma^\star$ denotes the set of finite-length strings over alphabet Γ and ϵ denotes the empty string. A PDA is formally defined as a 7-tuple [5]:

$P = (Q, \Sigma, \Gamma, q_0, Z, F, \delta)$

- Q is the set of states
- Σ is the set of input symbols
- Γ is the set of pushdown symbols (which can be pushed and popped from stack)
- q_0 is the initial state
- Z is the initial pushdown symbol (which is initially present in stack)
- F is the set of final states
- δ is a transition function which maps $Q \times \{\sigma \cup \epsilon\} \times \Gamma$ into $Q \times \Gamma^\star$.In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.

The change of state is determined by the transition function. There are two methods of acceptance for a pushdown automaton. They are final state acceptance where the automaton is considered completed when it reaches a certain final state or it could be accepted when the stack becomes empty. these two methods can be used interchangeably. There are importantly two kinds of PDA namely deterministic PDA and non-deterministic PDA.[5]

Deterministic PDA (DPDA) is a variation of the PDA where given a set of current state, input element and the top stack element there is only one state the automata can go to. Non-deterministic PDA (NPDA) is a variation of the PDA where given a set of current state, input element and the top stack element there are more than one state transition that are valid for this automata.

This paper follows the following thought process. We first prove that the business world in fact represents a chaotic system. This leads us to the fact that the business world depends largely on its initial conditions and hence we can use a pushdown automaton to represent the states of the enterprise. However, due to the dynamic nature of the business world using conservative methods to obtain business intelligence may not provide accurate results. This provokes us to explore a new paradigm which looks at history of decisions and their impact rather than conventional data. The next few sections will explain this paradigm extensively.

II. Formalising the System

Any enterprise needs to work with multiple other enterprises to sustain in the business world. This leads to the formation of a large interconnected network of companies where every enterprise is dependent on some other enterprise to fulfil parts of its needs. This makes it more complicated to determine the outcome of an event since there are a lot of factors at play. This dynamic system is largely dependent upon the conditions of the entity under considerations and the input it takes to determine the next state. This can be classified as chaotic system where the outcome of decisions is determined by the initial conditions of the enterprise and the input you provide to it. In today's fast paced economy, an enterprise should be capable of taking quick and accurate decisions. The enterprises need to strategise with conviction and take these decisions based on the larger picture that they have drawn for themselves.

Enterprises have realised that the only way to maintain the edge in this competitive and chaotic world is by building effective strategies. This might seem counter intuitive because we just spoke about the lack of predictability and yet strategy means we expect some things from our actions. However, an absence of a strategy is like floating on a ship in the middle of the ocean rather than steering in a direction, you are eventually going to run out of food. So the only way left for an enterprise is to strategise as best as possible by bearing in mind the multiple factors that affect your decisions. Since the business world is a complexly connected and highly dynamic system that represents chaotic behaviour, the principles of chaos theory might be of use to take decisions with conviction.

Chaos theory stresses on sensitivity to initial conditions which means that the route taken by an enterprise to achieve a certain status is invaluable as long as it is an ethical and legally binding route. The only situation that matters is the situation the enterprise is in now irrespective of the multiple situations it has been through. All the decisions should be taken by keeping in mind the current spot, but also keeping in mind the next spot the enterprise needs to attain. The next spot to attain is defined by the hypothesis of each of the options available. The quality of a bad decision is not defined by the hypothesis of the choice but by the failure to fulfil the hypothesis or due to the unexpected and uninvited ramifications it brings along. It is the duty of the decision maker to also consider the various ways the decision might affect the enterprise. The best place the consequence of a decision is seen is the stock market. Taking good decisions and living up to it always drives your stocks high.

The reasons the non-deterministic PDA is a good fit because it is capable of inculcating concepts of chaos theory and renders a fairly sensible output for business intelligence. The initial conditions are represented by the state the company is in right now. The state the company strategizes to attain is represented by the final state. The top of the stack adds more dimensions to the initial state. The inputs represent the decisions taken by the enterprise and the various ways it affects the enterprise. The transition function determines onto which state the company will jump based on its decision. However it is non-deterministic because it is impossible to foresee every single factor that will affect the enterprise. Hence there is a bit of an unpredictability factor that goes in the transition function that allows the system to move to different states even if it has the same factors.

The various factors that affect the business can be classified into Internal and external factors. The internal factors refer to anything within the company and under the control of the company no matter whether they are tangible or intangible. [2] The internal factors are:

1) Plans & Policies
2) Value Proposition
3) Human Resource
4) Financial and Marketing Resources
5) Corporate Image and brand equity
6) Plant/Machinery/Equipments (or you can say Physical assets)
7) Labour Management
8) Inter-personal Relationship with employees
9) Internal Technology Resources & Dependencies
10) Organizational structure or in some cases Code of Conduct
11) Quality and size of Infrastructure
12) Task Executions or Operations
13) Financial Forecast
14) The founders relationship and their decision making power.

On the contrary to internal factors, external elements are affecting factors outside and under no control of the company. The external factors are grouped into two types micro factors and macro factors. [2] They are:

Micro factors:

1) Customers
2) Input or Suppliers
3) Competitors
4) Public
5) Marketing & Media
6) Talent

Macro factors:

1) Economic
2) Political/legal
3) Technology
4) Social
5) Natural

The above list concludes with a number of factors that affect an enterprise; however the enterprise has control over the internal factors only. Each factor can be further studied about how they actually determine the fate of the enterprise. The situation of external factors is more or less fixed and more importantly cannot be controlled by the company. The enterprise is forced to cope with its changes by making shifts in their internal factors. This paper deals with internal factors and their implications on the enterprise. Hence we exclude the external factors from consideration in this paper.

Now that we have decided what all factors to study, we need to choose how we measure these factors. We need to find a quantifier that can quantify and say that this factor is heading in the right direction. This quantifier needs to prove that the value added by a certain factor because of a specific decision has contributed productively to the enterprise or has brought ill fate to the enterprise beyond a fraction of doubt. Finding such a quantifier might seem difficult but simplifying it and looking at it in a different perspective helps us conclude on such a quantifier. Any decision is said to be good if it adds value to the enterprise and a decision is said to have added value to the enterprise if because of it has indeed increased the value of the enterprise. This is determined by the stock value of the enterprise. The stock price of a company always goes up if a good decision is taken and goes down if a bad one is taken. This quantifier can be used for all the factors and acts as a common denominator for all the factors.

III. Building the paradigm

After choosing the factors and the quantifier we need to next choose how to represent it so we can find fruitful insights for better decisions. In order to achieve we need to overcome a few roadblocks. Each decision demands the factors to behave in a certain way collectively. Hence the requirements that needed to be studied might change based on decision under consideration. Another issue is if we choose to study the various factors collectively and the interactions between them, we arrive at a very complex system that will fail to make definitive sense. However, we eventually want to arrive at how to manage the factors so the enterprise eventually arrives at a good decision.

Since the complexity increases exponentially with every added factor we choose a single factor at a time for our study. Even though different decisions require the assessment of different factors all decisions will affect the internal factors in one or the other way. Most often the requirements that need analysis will be a combination of these factors. Hence studying all these factors individually can tell the company how each of them need to be handled to ensure good decisions. However there might be an argument that just because the stock prices went up after a decision taken on a certain factor does not mean it went up only because of it. We will soon resolve this issue as well.

A pushdown automaton is built for each factor. The initial state when this process is defined by q_0 and the stack we use here is a special stack that can hold tuples. Of course each unique tuple can be represented by a unique stack symbol, but for simplicity sake we will push and pop the tuples. We consider only one decision at a time for a particular factor. The tuple we choose to push/pop on the stack is a tuple of states of all the other factors and the corresponding stock price. The decision that taken for that affects the factor is represented by the input symbol.

The initialisation for this NPDA is done by pushing the states of all the other factors and the stock price at that point of time onto the empty stack.

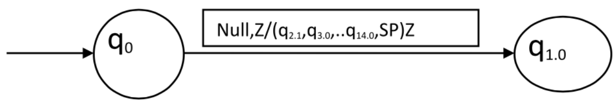

Fig. 1: Initialisation state where q1.0 represents the first state of factor 1, Z represents empty stack, Null represents no input

This consideration of input, the current state and the stack top variable leads to three legal transitions. They are as follows:

1) This is bound to bring about some changes in this factor and maybe others as well. In this scenario we change the state of the factor, we push the new changes in other states because of this decision and the corresponding new stock price after the change in state occurs onto the stack.
2) There are cases where the decision affects only the concerned department so there is a state change only in this factor but not other factors. In this scenario we push the same tuple and the corresponding stock price after the decision was taken onto the stack and then changes the state of the factor under consideration.
3) There are scenarios where the decision taken in one department does not affect that department but instead affects other departments. This means that the state of the factor under consideration will not change but instead states of other factors will change. In this scenario we circle back to the same state and push the changed state of other factors and the corresponding stock price after the decision is taken.

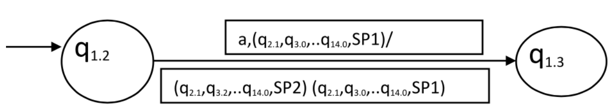

Fig. 2: change of state where multiple factors including itself are affected. 'a' represents the decision under consideration, SP represents a stock price

The scenario where a decision does not affect its own department or any other department is not considered a decision

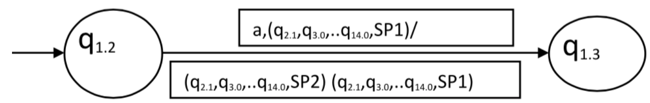

Fig. 3: change of state where no other factor but itself is affected.'a' represents the decision under consideration, SP represents a stock price

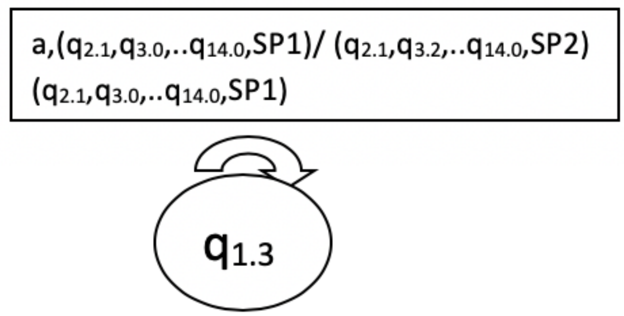

Fig. 4: looped change where other factors are affected but not itself

at all. An innate quality of any decision is the fact that it has implications. Hence there are only three possible scenarios possible. The above transitions rules are applied for a series of data until a stopping condition. This stopping condition varies depending on the purpose of the NPDA. If the NPDA is used for populating the stack, then its stopping condition will be until the desired data is empty. Generally data pertaining to stocks and other subsidiary data is taken within a historic time range. This data is analysed and used to populate the stack . If it is being used to build strategies then the final state of the NPDA is set based on the stock price the enterprise wants to achieve.

If you observe the above methods you will find that there is no pop method used. This means that ideally the stack needs to be infinitely large or should be large enough considering the number decisions that is under study.

IV. Procedure

The data for the above operations can be obtained by careful study of historic trends of the stock market and the decisions that was a precursor to it. However this data in its raw form is of no use for business intelligence. The steps involved to extract valuable information from this raw form is as follows:

1) Select data from various companies which have performed good and bad. A good mix of data from bad and good situations is necessary to classify both good and bad decisions. Make sure the selected data is not very

old because they tend not represent the current trends which leads to inaccurate results. Ensure that the data being analysed is not for a short period of time, because this will also lead to inconsistencies.

2) Pre-process the selected data from the source so it properly conforms to the requirements to the NPDA that we have modelled. This includes extracting each factor and analysing its individual properties and how a change in its properties can affect other properties. In this step we can subtract the stock price of the company under consideration before the decision from after the decision. So if the difference is positive, then it was a good decision else it was a bad decision.
3) Run the NPDA with the above pre-processed data. This is done for the entire data without partitioning it. This leads to populating all the stacks of all the factors for the different companies chosen.
4) Extract these populated stacks that are rich in data. They are in different silos based on the company and factors. Now it is important to integrate them based on the common factors. This will form a large data table with a deep data set of tuples from different companies.
5) Partition this data set into two other data sets where one has data which enabled the stock price to improve, while the other dataset contains data where the stock price eventually reduced. This will form two different data sets: one which contain consequences of good decisions and the other which is a consequence of bad decisions. Since it is a tuple which consists of a combination of the states we can say that the good decision is not because of just the situation of one factor but the collective state of all of them. This also ensures that we look at all the factors as a whole which seemed so complex at the beginning. If there are a few exceptions it is ignorable due to the large dataset under consideration.
6) Perform association rule learning algorithms such as Apriori with an appropriate support value and confidence value. Based on what the user is looking for, Sequential pattern mining can also be performed to obtain valuable insights. However you need to ensure that the sequential pattern mining is being performed on data from the same company as a sequence, or else it will lead to inaccurate results. Perform these algorithms on both the dataset.
7) Analyse these results obtained from the algorithms. The result from the good decisions dataset will tell you the combination factor properties that are key to success. The sequential pattern mining algorithm will tell you the series of decisions that were taken to obtain a winning combination of factors and ultimately achieve success. The result of the bad decisions dataset will tell us what a bad combination of factors is and the situations we need to avoid. It will also tell us what a series of bad decisions look like and help us recognise one when we make a mistake. These results can be used to better the enterprise in various areas.

V. Results and analysis

The results of the algorithms obtained on the datasets can give valuable insights on what consists of a good decision and a bad decision. It also helps companies find out if they are heading along the wrong path early in the process. This gives them a chance to better themselves and grow at a quicker rate. It also determines important combinations that are vital to success and help decision makers make important decisions quickly. This information can also be used to strategise and set targets. First setting a final state for their company's NPDA such as a combination of different factors being in certain states or a stock value, and then working backwards by building intermediate states with a timeline until you reach the present situation is a novel and credible approach to business intelligence.

As much as it is satisfactory to know the advantages of this method, there are drawbacks as well. This suggested model for business analysis does not include external factors. These external factors bring about a wave of unpredictability to the system. For instance if a stock price has reduced but the economy was bad and all competitors lost on the market that day, this reduces the credibility of the system. This can be improved to some extent by considering relative gain or loss rather than the absolute stock price. The relative gain or loss can be defined as the gain or loss of the company under consideration divided by the average gain or loss experienced by similar companies in the same time frame. If the relative gain is more than one, or relative loss less than one, then the decision is considered to be a good one, else it becomes a bad decision.

VI. Conclusion and future work

This paper talked about chaos theory, pushdown automata and the various concepts that help in business intelligence. It also showed a novel way of integrating these three concepts for building better business strategies. It then gave a procedure of how to obtain valuable insights from such a model and apply it in the real world. This method of obtaining intelligence can reduce the load on the large amounts of customer data stored in high end data warehouses for insights. This paper acts as building block in novel methods to obtain intelligence and provides groundwork for further improvement on this model.

References

[1] B. Marshall ,D. McDonald, H. Chen, W. Chung, "*EBizPort: collecting and analyzing business intelligence information*", JASIST, 55, p:873-891, Aug, 2004

[2] Mageplaza, " *What Are Internal & External Environmental Factors That Affect Business*", Nov. 29, 2017. Accessed on: Feb 26, 2020. [Online]. Available: https://www.mageplaza.com/blog/what-are-internal-external-environmental-factors-that-affect-business.html

[3] C Howson, "*Successful Business Intelligence: Unlock the Value of BI & Big Data*", Second Edition. New York,US: McGraw-Hill Osborne Media, 2013

[4] K.T. Alligood, T. Sauer, and J.A. Yorke, "*Chaos: an introduction to dynamical systems*" Berlin, Germany. Springer-Verlag, 1997

[5] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "*Introduction to Automata Theory, Languages, and Computation*", 3rd Edition, Boston, USA, Addison-Wesley Longman Publishing Co. Inc., 2006