Data-Centric Crisis Management

Serge Plata CMath CSci FIMA, Nomad Foods and **Sumanas Sarma**, Sainsbury's Argos

The main purpose of this article is to propose a tool for businesses that empowers decision makers: (a) to evaluate the risk of getting into a crisis before it has started; (b) to analyse different scenarios and the most likely outcomes during the course of the crisis and (c) to analyse the impact of the new collection of data in subsequent forecasts and evaluations once the crisis has passed. Note that the perception and interpretation of new data before, during and after the crisis is an epistemic problem in itself and therefore, has to be addressed separately. We will not discuss this here but rather concentrate on the use of machine learning techniques to approach this problem.

Introduction

First, we would like to clarify that we are not trying to say, by any means, that a crisis is a game or that it can be compared to a game or something playful. On the contrary, we take crises very seriously and that is why we have tried to implement mathematical models and computational solutions to deal with them more efficiently. When we say that we can compare a crisis to a game, we are doing so only so that we can apply a mathematical model that may be able to identify the best strategic solution to a game and by extension to a crisis.

Crises often occur in business, and with the COVID-19 pandemic, the analysis and management of crises have become more important. We have observed many ways of dealing with this pandemic, from lockdowns to doing nothing.

We started working on possible solutions for this specific crisis and noticed a number of parallels between crises and games:

1. *Limited data*. In a crisis, one has limited information when making decisions. In a game (like the prisoner's dilemma), we also have limited information.

2. *Action*. In a crisis, one has to act iteratively. In a game, one has to move or play.

3. *Other players*. A crisis can have many dynamic factors. In a game, the other players are making moves dynamically.

4. *Evaluation*. In a crisis, one needs to evaluate the situation at a given point and calculate the best options to get out of the situation. In a game like chess or go, one evaluates the board at a given point (after a number of moves) and calculates the best course of action for winning.

5. *Learn quickly*. In a crisis, one needs to learn quickly, as such situations are anomalous and one may not have much experience in dealing with them. In a game, one knows the rules and the allowed moves, and one has to learn quickly how to play.

So, we thought that we could model a crisis as a game. This could support decision makers in evaluating a situation with limited data and help them to figure out the best course of action among the many options. Moreover, using advanced computer methods, one can leverage the data available by incorporating the data into the model, even if the data are complex, convoluted or intricate.

A crisis as a game

In the 1940s, John von Neumann and Oskar Morgenstern published *Theory of Games and Economic Behavior* [1], which proposed a number of mathematical tools for modelling economic and general strategic situations as games. More or less ten years later, in the 1950s, we saw the application of computer science to an actual game, when the alpha-beta pruning method was used to optimise move evaluation in chess [2].

For many years, computer scientists and mathematicians worked on the problem, but meaningful progress has only been made recently, as we have (i) the computing power to evaluate the large number of possibilities and (ii) the mathematical background on neural networks to learn quickly how to play.

Our model is based on these premises.

What is a crisis?

The use of the word 'crisis' began in medicine as a decisive point in the development of a disease. We understand this as the possibility of an actual death. The word comes from the Latinised form of the Greek *krisis*, which, according to `www.etymonline.com/word/crisis`, means 'turning point in a disease, that change which indicates recovery or death'.

But we would like to give a more focused definition that applies better to generalised crises as we understand them outside the medical profession.

First approach: A crisis is a *situation* in which the possibility of *unintended termination* is high, in other words losing the game. This certainly extends to scenarios that include the loss of human life, hence the importance of approaching this from many angles. However, our work focuses exclusively on business situations. For example, say a company realises in week 1 that it is losing sales. They decide to adopt a price-matching policy, i.e. they will match the lowest price offered by their competitors to try to increase sales. A week later (week 2), the company is still losing sales. In this case, the situation might not be critical; they might be losing sales but the company is not in danger of termination. However, the following week after taking some action, the situation may become critical, as the termination state could be closer. In other words, losing sales can be bad but is not critical or terminal, and even a week after taking measures, losing sales is still not critical. It might even be really bad, but it will not reach a critical point, like a termination. So, is losing sales undesirable? Yes. Is losing sales unexpected? Yes. But is losing sales a crisis?

This business example can be extrapolated to any other situation where a crisis is difficult to define and act upon; including situations in which the state of crisis is subject to interpretation.

Second approach: The key words in our first approach were *situation* and *unintended termination*, so we would like to add a quantifier, in this case, a probability. The idea is to determine the probability of an unintended termination in a situation.

Calculating a probability in this case is the same as building a forecast, with all the inherent difficulties and subtleties of doing so, for example, short-term versus long-term forecasts, identifying the correct metrics, etc. Moreover, forecasting can be very

difficult, because when the situation turns critical (or into a crisis), the historical data, on which the forecast is based, will become irrelevant (unless a similar crisis has happened in the past).

In addition to all these complexities, there is another important factor: the agents are constantly taking actions. Take the example of the company losing sales. The analysis done to determine the probability of a termination may be totally irrelevant the following week when it has acted to match competitors' prices, since this changes the market and company's performance.

In this case, a forecasting model would be a fantastic tool for decision makers. However, a classic forecasting model may be unable to cope if the players are continually making moves.

Hence, our *third approach* incorporates *probability* and *players*. We realised that a crisis is analogous to a game, which has players constantly making decisions and playing. So, we thought of translating all these crisis situations and critical moves into a game frame. Although applying the classic theory of games developed by von Neumann is feasible, it might be difficult to implement due to the number of variables and the amount of data that needs to be processed iteratively in a very short time. Thus, we decided to apply a machine learning approach.

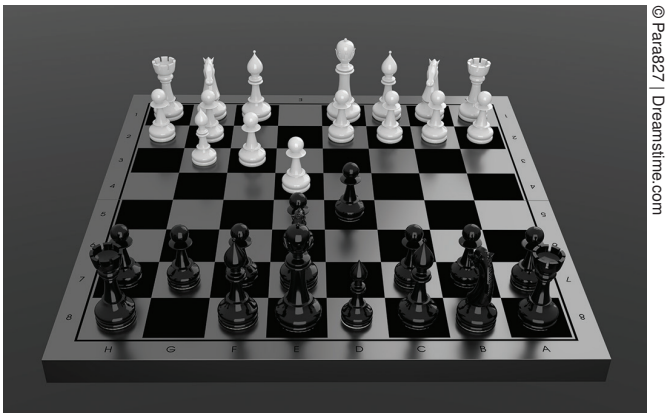

Figure 1: State of a chess game after some moves.

Thus, instead of using the term *situation* we will refer to a *state*. This is because states are widely used in games, so we will take advantage of game jargon to help us define business situations. For example, when playing chess (or go), a state of the game is a snapshot of the board at a given point of the game. The state of a company would be the set of all the variables at a given point of time. This state gives us a good picture of what the game board is after a number of moves or actions have been taken.

Now we can define a crisis in terms of a game: *A crisis is a state in which the unintended termination state is probable after a certain number of moves*.

Technical preliminaries

The main techniques applied in the classic theory are:

- Exhaustive searching, e.g. a minimax/alpha-beta or selective search
- Statistical sampling, e.g. a Monte-Carlo tree search
- Evolutionary algorithms

However, due to the number of possibilities that have to be explored, it is not feasible for even the most powerful computer to make an exhaustive search or a selective search, or even to apply a minimax algorithm. So, we took advantage of two breakthrough programs: AlphaGo and AlphaZero. One of the main concepts used is a neural network (p, v), where p is a probability and v is the value of the network at some point:

$$(p, v) = f_\theta(s), \tag{1}$$

where θ is a parameter of the network that is adjusted to optimise a loss function l that sums over the mean squared error and cross entropy losses:

$$l(z - v)^2 - p^T \log(P) + c|\theta|^2. \tag{2}$$

This neural network takes the state or board position s as an input and outputs a vector of move probabilities p with components $p_a = P(a|s)$ where a is an action and s is a state. The expected outcome of z given the position s is the vector $v = \mathbb{E}[z|s]$, see [3].

Our model, like AlphaZero, uses a Monte Carlo tree search (MCTS) algorithm to find the most promising action given a state or board position s. Each search consists of a series of simulated games of self-play that traverse a tree from an initial state s_0 until a leaf (or branch of the tree) is reached. Each simulation proceeds by selecting in each state s an action a with a low visit count, a high move probability and a high value according to the current neural network $f_\theta(s)$. The search returns a vector v representing a probability distribution over moves, $v_a = P(a|s_0)$ [3], with the more promising moves having larger probabilities.

Our model

Our model focuses on what we call the decisive horizon. Decisions are taken when the physical system is on the horizon and the possibilities are in the near future. We call it the decisive horizon as it is in the short term and there is no attempt to predict far into the future, since the many players are constantly changing the situation and conditions. This horizon encompasses a wide range of possible outcomes, which are classically included in a light cone [4]. We chose the light cone as it is like a tree that spans up from its roots and through all the branches and leaves. The possible states in the cone are the result of probable situations; in summary, our model is a set of stochastic processes analysed through a search function within a tree structure.

Another feature is that our model analyses many dimensions of a situation and is able to include as many conditions and variables as necessary. We call this method the conic search horizon multidimensional tree: COSHMERE.

The human making decisions will no longer be responsible for considering every possible outcome and the model will no longer be responsible for always recommending the ideal strategy. A key part of this relationship is for the human to be aware at all times of the information that is available to the model. After all, the model (like humans) cannot know what it does not know.

The ability to discover strategies using only a set of rules is powerful, but it becomes invaluable when the concept of a game state is expanded to cover not just a turn-based board game, but any scenario in which actions have consequences.

It is important to note that the key assumption behind both AlphaZero and AlphaGo make them unsuitable for replacing human experts: they expect that all information is always available and that one player's gain is directly equivalent to the other player's loss. However, a human subject-matter expert can weigh the options proffered by the algorithm before making a final decision.

The key questions become: (a) Is it possible to come up with a set of rules for a business? (b) Given the set of rules and a scenario (treated as a game state), is it possible to objectively compare the state against another (essentially an objective function)? (c) Can we design a simulator that provides vast amounts of simulated data given the previous two rules? If these three criteria can be met, then a tabula rasa learning algorithm, like AlphaZero, can enhance the decision-making of a human expert during a crisis.

Neural networks

Neural networks can be considered to be universal function approximators. Their ability to estimate functions has been explored in the literature [5].

If we assume that there is a function that can accurately estimate the value of an *action* in a given state, then a neural network can provide a suitable approximation of this function. If a small change is made to the parameters of this function (for parameters that have a known result), then the neural network, trained appropriately, can provide a result within a certain margin of error.

A classic example is the task of learning the function that returns the square of its input. Once trained, the neural network can generalise to input values that were not provided during the training. This ability to generalise is crucial, if there is an underlying assumption that the patterns in the data are consistent. Naturally, during a rapidly changing situation, it is unreasonable to require that the data patterns remain consistent with those that existed before the situation. In retail, the COVID-19 pandemic has significantly changed buying patterns in a manner that makes any comparison to previous annual data troublesome. For such scenarios, the role of the human decision maker is crucial, until the network is able to generalise from new patterns in the data. A symbiotic relationship between the model and the human is ideal, as each will bolster the other's weaknesses.

Penalty and reward functions

In supervised learning, if the output of the model has a consequence, then we have the necessary conditions for reinforcement learning. A simple image classifier that makes idempotent classifications is not an example of such a model. However, a model that can play a game such as chess or go is, because the possibilities change during the game depending on the moves made.

The fundamental learning step is feeding the correct or incorrect result back to the model. Penalty and reward functions can tell the algorithm to pay more attention to certain variables. The goal of penalty functions, in general, is to convert a constrained problem (like a linear programming problem) into an unconstrained problem by introducing a penalty for surpassing the constraint.

When we apply a reward or penalty function in MCTS, the neural network makes the decisions. Instead of using brute force and exhaustively exploring all the possibilities, the decisions made at a given leaf of the tree are rewarded or penalised, and that is how the search is optimised.

Monte Carlo tree search

Monte Carlo trees are based on the same principle as the Monte Carlo simulation method originally proposed in 1947. The Monte Carlo method is a statistical sampling technique that has been applied successfully to many scientific problems. MCTS is used to search the most promising nodes when it is not possible to search all possible nodes. Each move can be represented in a tree form where the decisions taken follow a branch or leaf of the tree.

In a brute force approach, there is an exponential growth in the number of possibilities to explore, which requires significant computational power. Most of the time, it is extremely slow to compute the entire set of combinations. For example, in a binary decision problem like the knapsack problem, the number of calculations for only 20 items is $2^{20} = 1\,048\,576$.

That is why a faster search can be achieved with a policy, i.e. by giving more importance to some nodes than others based on a value function so that their branches have a higher priority of being searched. MCTS can figure out the most promising move out of a set of moves.

Algorithm

At the heart of any turn-based game is a set of states and a tree structure waiting to be explored. In most games (like chess, go and shogi), the vast majority of these states will never be fully explored. Whilst this can be seen as counter-intuitive, considering only the most promising strategy (at the time) is something we humans do almost intuitively. The difference here is the ability to quantitatively compare and contrast a significantly larger number of possible strategies without being affected by any human biases (either conscious or subconscious) or assumptions.

The number of possible states that can be reached from a given board position grows exponentially with time. Roughly, this can be seen as the number of legal moves per position raised to the power of the game length [6]. This makes an exhaustive search of all possibilities computationally impossible, for even just a few steps into the future. However, by employing an estimation function in the form of a neural network, it is possible to make some probabilistic assumptions about the result of taking certain actions from a given state. This approach of running a large number of simulations from a given state based on the premise of each possible outcome is a standard Monte Carlo simulation approach. Using a game state navigation model to make recommendations can allow humans to make probability-informed decisions about the next steps to take during a crisis.

Further developments

We propose to implement and productionise our model in companies to prevent and forecast possible crises. Moreover, during a crisis, the COSHMERE model can recommend possible solutions and evaluate quantitatively the risk of an unintended termination state and the most promising actions.

References

1 von Neumann, J. and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*, Princeton University Press.

2 Knuth, D. and Moore, R. (1975) An analysis of alpha-beta pruning, *Artif. Intell.*, vol. 6, pp. 293–326.

3 Silver, D. et al. (2018) A general reinforcement learning algorithm that masters shogi, and go through self-play, *Sci. Mag.*, vol. 362, pp. 1140–1144.

4 Plata, S. (2007) *Visions of Applied Mathematics*, Peter Lang, Oxford.

5 Gurney, K. (1997) *An Introduction to Neural Networks*, UCL Press, London.

6 Silver, D. et al. (2016) Mastering the game of go with deep neural networks and tree search, *Nature*, vol. 529, p. 484–489.