Reinforcement Learning: AI that creates AI

Atul Singh, Vishal Kumar and Shashank Hegde

Fidelity Investments Bangalore

Abstract: Selecting the supplier that can provide the desired products and services while fulfilling the operational constraints is key to the success of an enterprise. Typical supervised machine learning algorithms can be used to predict the ability of a supplier to satisfy the operational constraints such as time to deliver and cost. Optimization algorithms can be applied on the predictions to select one or more suppliers to achieve the best performance on the operational constraints. Once in production the lack of human intervention means the supervised machine learning algorithms will not get the latest labeled data required to train the model for accurate predictions. Deep Reinforcement learning merges the progress made by deep neural networks with reinforcement learning and has shown remarkable success in literature. The ability of deep reinforcement learning based algorithms to explore, and exploit ensure that they do not suffer from the drawbacks of typical machine learning based algorithms in production. This paper presents an overview of deep reinforcement learning techniques and a deep reinforcement learning based solution for supplier selection.

Keywords: Deep Learning, Reinforcement Learning, Supervised Machine Learning, Supplier Selection

1. Introduction

Adam Smith, the father of modern economics attributes division of labor as a primary cause for improvement in production in modern organizations. Division of labor entails that organizations use the products and services provided by external suppliers for functions that are not core to their line of work. Selecting the external suppliers for products and services is critical to the success of an organization. The supplier selections problem exists in all the domains including but not limited to finance, manufacturing and telecom. Existing work typically uses Analytical Hierarchy Process (AHP) to identify the key criteria for identifying a good supplier [1]. The criteria are used to rank a supplier using historic data. The existing approaches typically do not take into account the varying macro conditions under which the suppliers operate, and which may impact their ability to fulfill the order while satisfying the operational criteria determined by the AHP process.

This paper explores the use of machine learning to select one or more suppliers given the macro conditions and the order attributes so that the performance of the order on a given criteria is optimized. The criteria could be the amount of time required or cost to fulfill the supply.

The order attributes would vary with the domain and will include the quantity and the type of the desired product and services. Please note that the ability of a supplier to fulfil an order may vary with the quantity and the type of the product. A supplier may have the capability to supply spare part in thousands but when the required quantity is increased then he might not have the capacity to fulfil the request. Let us look at examples of macro conditions that may impact the ability of a supplier to fulfil the order. For a supplier in the manufacturing domain the amount of raw materials produced in the year, the market labor supply, and the market labor rates may impact the ability to fulfill an order. A supplier in the financial domain who fulfills equity orders from the market will depend on the stock market conditions such as liquidity and volatility to fulfill an equity order.

Supervised machine learning techniques such as decision trees, linear regression and random forest can be trained using labelled examples to create a machine learning model that can be used to predict the performance of a supplier on a given criteria [2]. For our supplier selection problem, the labelled examples contain the different macro environment attributes along with the order attributes as the input, and the known performance of the supplier for the macro environment and order attributes as the labels. The supervised machine learning model uses the labelled training examples to learn a function that can be used to predict the performance of the supplier for a new macro environment and order attributes. The results of the supervised machine learning model can be used in an Integer Linear Programming (ILP) optimization technique to select the best suppliers for the order.

Bounded rationality is the idea that “when individuals make decisions, their rationality is limited by the tractability of the decision problem, the cognitive limitations of their minds, and the time available to make the decision”. The challenge with supervised machine learning algorithms is that the quality of the labelled data is limited by the bounded rationality of the human agent. This means that apart from normal human errors in labelling due to carelessness, the labels might also not reflect the best prediction because a supplier has not been tried for the given scenario. Furthermore, once the supervised machine learning model is deployed in production without human intervention then we will not get the labelled data required to continuously retrain the model to reflect the changes in the abilities of the suppliers to fulfill an order.

Reinforcement learning [3] is another family of machine learning algorithms that do not depends on labelled data for learning thereby removing the aforementioned problems with supervised learning caused due to quality or absence of labelled data. Like many great ideas in the field of computer science reinforcement learning also draws inspiration from nature where a young human brain explores and learns to master her environment often without any manual or historical information. Reinforcement learning is the process of learning by interacting with an environment through feedback. This paper presents a solution using reinforcement learning for supplier selection problem to select the suppliers given the macro conditions and the order attributes so that the performance of the order on a given criteria is optimized.

The rest of the paper is arranged as follows. First, we present an overview of the reinforcement learning algorithms. Next, we describe the details of the reinforcement learning algorithm used by us to solve the problem of supplier selection. After that we present the simulation results for applying the reinforcement learning algorithm on a dummy supplier selection problem. Finally, we present the conclusion of this work.

2. An overview of Deep Reinforcement Learning

Reinforcement Learning is a class of algorithms that addresses the problems with supervised machine learning by learning from the continuous response (also called reward) received by interacting with the environment. Reinforcement learning algorithms are also referred as reinforcement learning agents in this document. Reinforcement Learning algorithms interact with the environment through actions. They receive the observations about the environment and a reward from the environment for their actions. This information is used to decide the next action. Reinforcement learning has received a lot of attention recently because of the spectacular success of AlphaGo a reinforcement learning based algorithm for playing go over eighteen-time world champion Leo Sedol.

Markov Decision Process (MDP) is a theoretical framework that can be used to capture a reinforcement learning problem. MDP requires the following five pieces of information: set of states, set of actions, state transition probability matrix, immediate reward matrix, and a discounting factor. Reinforcement learning based algorithms use either a) a **policy function** that maps a state to an action, b) or an **action value function** called the q function that returns the expected reward from a given state for a given action. The policy function or the value function can be used to decide the actions that an agent should take at a given state. If all the information for a MDP are available, then the optimal policy that maximizes the rewards can be obtained by using dynamic programming-based techniques called policy iteration algorithms or value iteration algorithms.

But for most real-life problems such as supplier selection problem, all the information required by the MDP is not available. In such a scenario the reinforcement learning algorithms fall in two broad categories. The first category of algorithm estimates a policy function and the other category estimates a value function. The algorithms that estimate the policy function use the "gradients of the rewards with regards to the policy parameters, then tweaking these parameters by following the gradient toward higher rewards (gradient ascent)" [4]. TD-Lambda and Q-learning are popular methods that estimate the value function. The techniques for estimating value function typically use random walks from initial states to estimate the rewards for taking a given action from a state. Subsequently the policy function **q** for taking an action(**a)** from a state(**s)** can be estimated using the equation below:

$$q_{estimated}(s, a) = r\,(s, a) + \gamma * \operatorname{argmax}_{a \in A}[q_{estimated}\,(s_{next}, a)] \tag{1}$$

Deep Learning Neural Networks have shown immense potential in learning using labelled data. Deep learning neural networks can do a good job of learning from labelled data without the need of human driven feature engineering. This makes them a good candidate technique for estimating the policy function or the value function when the entire MDP details are not available. Marriage of deep learning neural networks with reinforcement learning has led to the emerging field of Deep Reinforcement Learning which has shown immense potential and had led to the algorithms behind the successful GO program AlphaGo. Techniques where deep neural networks is used to estimate the value of q function using labelled state and action data are called Deep-Q learning based techniques. Techniques that use deep neural networking to estimate the policy function are called the Deep policy networks. Deep-Q learning based algorithms don't scale well with an increase in number of actions. Furthermore, both the Deep-Q, and deep policy networks have challenges in converging to a right estimate of the value or the policy function.

Actor Critic model [5] is an enhancement that does not suffers from the drawbacks of Deep-Q and deep policy networks. The algorithm consists of two components: a) a critic that emulates a parent teaching a child how to drive a bicycle by constantly giving feedback, and b) an actor that does the real learning like a human child. The actor simulates a policy function that suggests the action that should be taken in a given state. The critic implements a value function that provides the maximum reward for a given state from a given action.

The actor takes as its input the observations about the environment and returns the action that the agent should take. The critic takes as input the reward generated by the environment and the observations about the environment and returns the maximum reward for the input parameters. The difference in the reward values obtained from the environment and the critic is used to retrain the actor. This actor critic model is also called Deep Deterministic Policy Gradients (DDPG) .

3. Using Actor-Critic model for supplier selection

We have implemented a modified version of the DDPG algorithm which uses only two representative input variables. The DDPG can be easily extended to accommodate more input parameters. The parameters defining the state include a) the total order quantity to be split among the different suppliers, and b) another variable representing the macro-environment variables. The state is fed to the actor network whose final layer is a softmax layer. It outputs the optimal split prediction, which we consider as the action. The state and action as defined above, are fed to the critic network to obtain the Q value for the given state-action pair. Considering the fact that the process of splitting is a two-step MDP, the Q value is considered equal to the immediate reward.

We maintain a buffer of a specific size, which contains tuples of the form (state, action, reward). For a given state, we predict an optimal split, and observe the total reward obtained by dividing the input quantity according to the predicted split and placing to the different brokers.

For training the critic network, we sample a random mini batch from the buffer and minimize the loss function which is the sum of the difference of the actual reward obtained and the Q value predicted by the critic network for all tuples in the mini batch.

$$L = \frac{1}{N} \sum_i \quad (y_i - Q(s_i, a_i | \theta^Q))^2 \qquad (2)$$

The actor is updated by the sampled policy gradient i.e the gradient of predicted Q value w.r.t the policy.

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \quad \nabla_a Q(\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i} \qquad (3)$$

4. Experimental Setup

For simulation, we decided to model the rewards of three suppliers as Normal distributions, each with a different mean, with respect to the quantity of trades assigned to it.

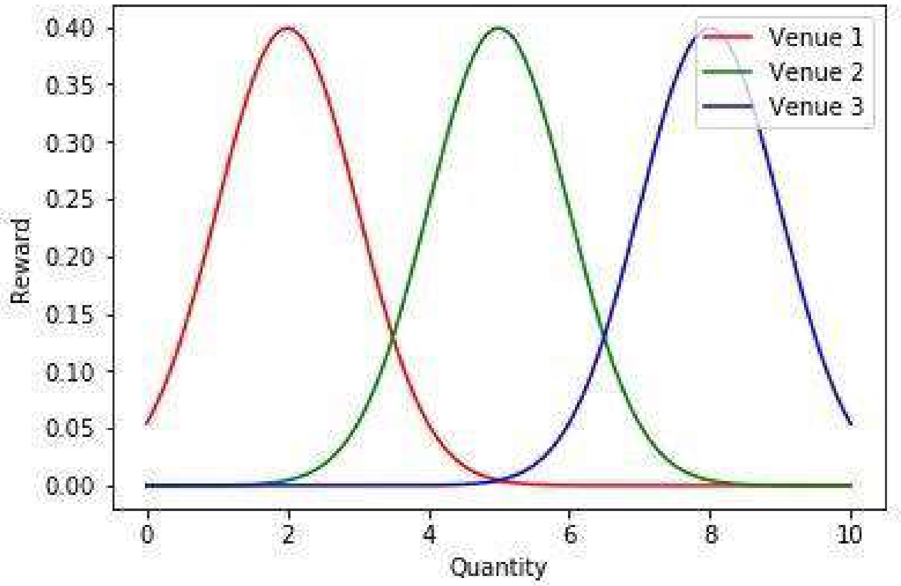

Fig. 1. Rewards at each venue. This figure depicts reward vs quantity for each venue. For example, Venue 1 (*red*) gives a maximum reward at quantity *2*.

It is important to note here that these rewards are unknown to the DDPG model prior to simulation. Simulation was done for each episode with the following steps:

- Feeding the DDPG model with a random order quantity and macro environment attribute.
- The model would predict a split for the entered amount.
- The Split was then sent to the environment.
- Based on how much quantity each venue was assigned, reward from each venue is summed up and returned to the DDPG model as Environment reward using Figure 1.
- The initial state (total order quantity, and the macro environment attribute), action taken (split predicted by the model) and the reward obtained is then stored in the replay buffer.
- Train the model.
- Go to next episode.

The above simulation is carried on for a large number of episodes. Each episode the model is trained.

5. Results

After running the simulation, the Actor network of the DDPG was tested on random Order quantities. Based on the Normal distributions used for each venue, the optimal split was calculated for these order quantities. The split predicted by the Actor was very close to the optimal split. The table below (Table 1) shows a test example.

Table 1. Comparison between Optimal split and Predicted split by the predictor.

Volume	**Optimal split**			**Predicted split**		
	Venue 1	Venue 2	Venue 3	Venue 1	Venue 2	Venue 3
15	2	5	8	2.34	4.93	7.72
2	2	0	0	1.8	0.096	0.104

The environment reward for the predicted split was also approximately close to the optimal reward possible for the entered Order. The reward obtained from the predicted split and the optimum reward had a Correlation of **0.71**. Since the Critic network was trained to predict the reward for a given Split, we were also able to estimate the maximum reward possible for an entered Order quantity. The reward predicted by the Critic and the Maximum reward possible had a Correlation of **0.83**.

6. Conclusion

This paper presents the supplier selection problem, and an overview of the reinforcement learning domain. It explores an actor-critic based algorithm for solving the supplier selection problem, and presents simple simulation results to validate the approach. Future work would include expanding the simulations to include more input variables.

Disclaimer

The views or opinions expressed in this paper are solely those of the author and do not necessarily represent those of Fidelity Investments. This research does not reflect in any way procedures, processes or policies of operations within Fidelity.

7. References

1. Chai, J., Liu, J. N., & Ngai, E. W. (2013). Application of decision-making techniques in supplier selection: A systematic review of literature. Expert Systems with Applications, 40(10), 3872-3885.

2. Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, pp. 337-387). New York: Springer series in statistics.

3. Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1). Cambridge: MIT press.

4. Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".

5. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.