Resource Allocation using Decision Focused Learning

Qazi Zarif Ul Islam

November 26, 2024

1 Abstract

2 Introduction

"Given some input data, output data and a feasible input space, determine the optimal combination of the input data for the optimal output data"- This is the problem we seek to solve. For example, Suppose we don't know the exact profit per unit for Product A and Product B, but we have historical data on production quantities and the corresponding total profits. Our goal is to determine the optimal production quantities of Product A and Product B to maximize profit, given resource constraints and without explicit knowledge of per-unit profits.
We will argue in this paper that the above task requires techniques beyond commonly used data driven techniques such as Machine Learning and statistics and finally propose a simple framework that tries to solve such a problem. The paper is structured in the following manner: section 2.1 discusses the central tenets of machine learning. Section 2.2 Discusses techniques that currently exist for problems that involve decision making using data.

2.1 The central tenets of Machine Learning

Machine learning (ML) is a field that primarily focuses on predicting some outcome, given some data. The central tenet of using such a framework is that the input variables that are used are somehow correlated with the outcome. For example, in the popular iris dataset, the input variables are the sepal and petal characterisitcs and the outcome is one of three species of the iris flower. The secondary tenet is that no observation made in this world is by itself the true representation of what is being observed and so *all observations* are noisy (contain noisy information). Thus, it follows that a single prediction made from a single noisy observation is also noisy (Noisy input gives noisy output). However, time and time again, such predictions have proven helpful. For example, a $\geq 90\%$ accurate transformer is deemed capable enough to complete incomplete passages or poems, computer vision systems are used to identify objects, understand human sentiments, etc. These wonders are due to the fact that all phenomena are generated by probability distributions and a random sample approaches a limiting probability distribution as the size of the sample increases. Thus, even though the learner is unaware of the underlying limiting distribution, as we increase the sample size, the particular error-correction algorithm of the learner leads it towards the true prediction.

2.2 Techniques that drive decision making

ML based products have been used primarily to make tasks more efficient, not to induce social policy and decision. We trust qualitative arguments when deciding where to build a hospital or where to put more health care funding. One explanation for this is that we can determine biases in an argument instantly and doing the same for a learner is much harder, simply due to the secondary tenet (see 2.1). Another reason for the insuitability of ML techniques in this area is that the predictions do not take into account *its own effects* on downstream decisions. But where we have not trusted predictions, we have trusted optimization and statistical testing. **Optimization techniques** have been heavily relied on when a deterministic answer is deemed to exist. Examples of such problems include The Travelling Salesman proble, The Knapsack problem, etc. Optimization problems differ from Machine Learning problems in that the problem is not solved using data but rather a fixed set of conditions. This fixed set contains a function relating the inputs to the output and equations that give the 'feasible answer region'. But, there are numerous problems where we don't have a function that gives the input-output relationship but yet we want to optimize the inputs for either maximizing or minimizing the output based on some constraints on the inputs. For example, the problem of socio-economic resource allocation in order to lessen viral infections. Another area that contains techniques that lead to decisions from data as opposed to predictions from data is **statistical hypothesis testing**. However, using hypothesis testing requires knowledge of the underlying probability distributions and real world data does not explicitly express the probability distribution it is generated from. **Data driven control** provides a way to *control* the input variables and thereby, optimize them for a desired output however, requires knowledge of the input-output relationships. Furthermore, data driven control is usually applied to dynamic systems where the variables describing the system changes with time. Time is indeed an important consideration in many problems but real world data often either describe a static system or the data pertains to the system at a particular timepoint.

The above limitations called for new techniques that can optimize inputs simply using historical data, by predicting the relationship between the inputs and ouput and then using the predicted relationship to formulate an optimization problem. [2] proposed the name "**Predict, then optimize**" for this framework and used the term '**Decision Focused Learning**' to describe the techniques. To our knowledge, [1] was the first to propose such a method, wherein a trained predictive model was used to predict a cost vector, given an input vector, and then the cost vector and input vector were used to form an objective function for an optimizer to solve. The cost vector $\mathbf{c}$ is basically the coefficient vector, which, combines with input vector $\mathbf{f}$ as follows to form an objective function f,

$$f(\mathbf{x}, \mathbf{c}) = \mathbf{x}.\mathbf{c} \tag{1}$$

The predictive model needs to be trained in a supervised manner and so, we must bear knowledge of how each variable in the input vector $\mathbf{x}$ individually affects the function f, which is not available in many real-world scenarios. We can use a more simplified approach wherein we are given input-output pairs, $(\mathbf{x}, \mathbf{y})$ and we have to produce a symbolic function, expressing the mapping between the input and output. The following section discusses such techniques in the existing literature.

3 Preliminaries

Suppose the goal is to determine a combination of inputs $\{x\}_n$, constrained by some conditions , that maximizes or minimizes an output, which is a function of $\{x\}_n$. This problem can be described as,

$$\begin{aligned}
\text{Given,}& \qquad (2)\\
f(\mathbf{x}) &= \mathbf{c} \cdot \mathbf{x}, \qquad (3)\\
g(\mathbf{x}) &\geq 0, \qquad (4)\\
h(\mathbf{x}) &= 0 \qquad (5)\\
\text{Determine,}& \qquad (6)\\
\mathbf{x}^* &= \arg\min_{\mathbf{x}} f(\mathbf{x}), \qquad (7)
\end{aligned}$$

Where $\mathbf{c}$ is the coefficient vector that is unknown. Fortunately, ML techniques provide a way to estimate this vector from data. If $\hat{\mathbf{c}}$ is the predicted coefficient vector, the problem becomes,

$$\text{Given,} \tag{8}$$
$$f(\mathbf{x}, \hat{\mathbf{c}}) = \hat{\mathbf{c}} \cdot \mathbf{x}, \tag{9}$$
$$g(\mathbf{x}) \geq 0, \tag{10}$$
$$h(\mathbf{x}) = 0 \tag{11}$$
$$\text{Determine,} \tag{12}$$
$$\mathbf{x}^*(\hat{\mathbf{c}}) = \arg\min_{\mathbf{x}} f(\mathbf{x}, \hat{\mathbf{c}}), \tag{13}$$

Where $\mathbf{x}^*$ is a function of $\hat{\mathbf{c}}$ because it is dependent on the prediction made by the learner. There are two ways of approaching this problem: (1) Modifying the ML learner so that it incorporates the optimization problem into its learning algorithm or, (2) Predicting and then, optimizing, which follows the method in [1].

3.1 Approach 1

In this approach, we don't care about the accuracy of the predictions, $\hat{\mathbf{c}}$, but rather how accurate the decision *based on* the prediction is ($\mathbf{x}^*(\hat{\mathbf{c}})$), which we call *Regret*. The Regret is formally expressed below.

$$Regret(\mathbf{x}^*(\hat{\mathbf{c}})) = f(\mathbf{x}^*(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^*(\mathbf{c}), \mathbf{c}) \tag{14}$$

Where $f(\mathbf{x}^*(\mathbf{c}), \mathbf{c})$ is the decision that would have been obtained had the optimizer exact knowledge of $\mathbf{c}$. $f(\mathbf{x}^*(\mathbf{c}), \mathbf{c})$ is known as 'full information decision'.

3.2 What are our options?

1. Gaussian mixture modelling and then hypothesis testing.

2. Data driven control- System identification +

3. Physics informed Neural Network

4. Miscellaneous

4 Conclusions

We need to find a way to automatically discover relationships between the variables themselves to be able to describe the dynamics of the system.