ECE 8725 Final Project Report

Connecting Data Driven Optimization, Decision focused learning and Symbolic Regression

Qazi Zarif Ul Islam

December 13, 2024

1 Abstract

Optimization problems often require finding the best combination of input variables to achieve desired outcomes under given constraints. While traditional Machine Learning (ML) techniques excel at predictive tasks, they often fail to consider the downstream impact of their predictions on decisions. Conversely, decision-focused techniques, which perturb inputs to achieve specific outcomes, lack the ability to leverage observational data. This report first discusses traditional methods of optimization without data and then with data and proposes a simple yet novel framework, **Express Symbolically then Optimize (ESO)**, which integrates symbolic regression and optimization to derive interpretable input-output relationships and subsequently optimize decision-making.

We discuss existing methods such as Stochastic Programming, Decision-Focused Learning (DFL), and Data-Driven Control, highlighting their limitations in addressing problems where explicit input-output functions are unavailable. The ESO framework addresses these limitations by using symbolic regression to uncover interpretable mathematical relationships from data, followed by optimization based on these symbolic expressions.

We conclude with some remarks about evaluating applications of this framework, which we believe will require novel methods that use post-hoc analysis of predictive models.

2 Introduction

"*Given some input data, output data, and a feasible input space, determine the optimal combination of input data to achieve the desired output*"—this is the fundamental problem we seek to address.

For instance, consider a scenario where the per-unit profit for Product A and Product B is unknown, but historical data on production quantities and corresponding total profits is available. The task is to determine the optimal production quantities of Product A and Product B to maximize profit, given resource constraints, without explicit knowledge of per-unit profits.

The above problem is an adaptation of the conventional optimization problem where a problem is provided in the form of a function, which is to be minimized or maximized. Only this time, the functional form is not provided and instead, we are provided *observations* about the inputs and the resulting output. Machine Learning (ML) and statistical methods have achieved remarkable success in prediction and inference with such historical data. But predictive models do not account for how the predictions affect downstream decisions or outcomes. On the other hand, techniques that focus on permuting inputs to achieve certain outcomes- henceforth called 'Decision Focused techniques'- do not leverage observations. Thus, we think it is important to bridge this gap by using principles from both branches- Machine Learning and Decision Focused techniques to tackle the problem at hand.

The rest of the paper is structured as follows. Section 3 discusses the background of Decision Focused techniques and promising techniques from Machine learning that can be used to bridge the gap. Section 4 formalizes the problem and briefly discusses how to tackle the problem with an emerging area in Machine learning, **Decision Focused Learning** and finally how to tackle it with the new proposed framework, *Express Symbolically, then Optimize.*

3 Background

ML techniques have been used primarily to make tasks more efficient, not to induce social policy and decision [1]. 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 as there are different kinds of biases and a plethora of experimental methods to discern them [9]. 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 problem, The Knapsack problem, etc. Apart from **stochastic programming**, a particular paradigm of optimization method, these 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, one could collected data on healthcare funding, number of different kinds of social establishments, poverty rates and disease rates and may want to decide if more socio-economic development is urgently needed or not. In this scenario, there is no known function to optimize as the relationship between the socio-economic factors and the disease rates is unknown. To solve this kind of problem, **stochastic programming** tries to use a 2 stage approach where in stage 1, decisions are taken in an outcome agnostic manner and in stage 2, the costs or benefits of these decisions are evaluated in terms of a probabilistic objective, typically the expected value of a performance function that incorporates randomness.. In this way, stochastic programming effectively models uncertainty by leveraging historical data or assumed distributions. The objective function is not directly "learned" in a machine learning sense but rather constructed to incorporate uncertainty and minimize the expected cost (or maximize expected benefit). **Genetic algorithms** or Evolutionary Algorithms use search algorithms inspired by population behaviours that exist in nature to solve optimization problems [3]. Just as in the previous methods, the objective function needs to be fully defined. Genetic algorithms truly shine when minimizing functions that are extremely complex, containing many local minima/maxima, for example the Rosenbrock or Rastrigin function [12].
Another area that contains techniques that lead to decisions 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. When this relationship is unknown, **system identification** is used to derive it. Data driven control and system identification are usually applied to dynamic systems where the variables describing the system changes with time. It is however possible to use another variable as a proxy for time to hold information about how the system changes with that proxy variable. First, we need to have a model structure, for example,

$$y(k) + ay(k-1) = bu(k) \tag{1}$$

Where **a** and **b** are adjustable (learnable) parameters. From this model structure, we use historical (input, output) data to derive the learnable parameters. Usually, the input and output data are recorded through time. If we instead have static data, though it is possible to simply index every observation and then use the indices as k, when the data is shuffled, it changes the dynamical system itself. Thus every shuffling order of the data gives a new dynamic system. Besides, there is no inherent temporal information in the indices, they are simply numbers with no physical or temporal meaning.

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. [6] proposed the name "**Predict, then optimize**" for this framework and used the term '**Decision Focused Learning**' (DFL) to describe the techniques. To our knowledge, [2] 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 simplest representation of such an objective function is,

$$f(\mathbf{x}, \mathbf{c}) = \mathbf{x}.\mathbf{c} \tag{2}$$

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. To our knowledge, [8], was the first to apply DFL in a real-world scenario to optimize the scheduling of live service calls in a maternal and child health awareness program. They used socio-demographic features (e.g., age, income, education) and historical engagement states (e.g., whether beneficiaries listened to calls) as inputs, while the engagement outcomes (e.g., listening behavior) served as outputs. The model predicted transition probabilities between engagement states (engaging/non-engaging) under different actions (call/no call). These predicted probabilities were used to compute Whittle Indices [11], which prioritized beneficiaries for live service calls based on their potential to benefit.

Decision Focused Learning incorporates the decision making into the learning algorithm (typically Backpropagation with Neural Networks). The function of the decision, $f(\mathbf{x}^*(\hat{\mathbf{c}}))$, is often non-differentiable and this poses significant challenges in applying DFL [6]. 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. Thus, in the end, the problem is simply one of **symbolic regression** (SR). After the symbolic expression of the function is obtained via symbolic regression, we use it as the objective function. Symbolic regression has been used to add a layer of 'interpretability' to black box machine learning methods that learn functions. We start with a set of possible arithmetic operators known as the 'library' and define the dimensionality of the function space. For example, a library could be $L = \{id(.), add(.,.), sub(.,.), mul(.,.), +1, -1\}$. This particular

library can compose the set of all polynomials in one variable with integer coefficients [5]. [5] conducted a comprehensive survey on SR methods and divided the techniques into 4 fundamental types; (1) Regression based methods (2) Tree based methods (3) Physics inspired methods and (4) Mathematics inspired methods. Genetic algorithms can actually be used as a tree based method for SR. Recently, **Kolmogorov Arnold Networks** (KANs) were shown to perform better at Symbolic Regression than any other method [4] [13]. Kolmogorov Arnold networks have the same skeleton as **Neural Networks** [10], but the difference is KANs use learnable functions instead of real valued weights as the edges of the computational graph. The vertices of the graph are simply summing operators. The learnable activation functions is the main ingredient of KANs, which are spline functions. This automatically produces the symbolic nature of the learned function. In Neural Networks, this symbolic extraction is not as easy. Thus, we propose using KANs to symbolically regress data and learn the function that describes the relation between the inputs and output. Then using the learned function as the objective function, solve the optimization problem.
In the following section. We discuss two approaches that we think could be adopted to solve resource allocation problems based on historical data.

4 Formalization of the problem

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,

$$\text{Given,} \tag{3}$$
$$f(\mathbf{x}) = \frac{1}{2}.\mathbf{x}^T.H.\mathbf{x} + \mathbf{c^T}.\mathbf{x}, \tag{4}$$
$$g(\mathbf{x}) \geq 0, \tag{5}$$
$$h(\mathbf{x}) = 0 \tag{6}$$
$$\text{Determine,} \tag{7}$$
$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x}), \tag{8}$$

Where H and $\mathbf{c}$ represent coefficients and are unknown. Fortunately, ML techniques provide a way to estimate these matrices from data. If $\hat{\mathbf{H}}$ and $\hat{\mathbf{c}}$ are the predicted coefficients, the problem becomes a quadratic programming problem like below.

$$
\begin{aligned}
&\text{Given,} && (9)\\
&f(\mathbf{x}) = \frac{1}{2}.\mathbf{x}^T.\hat{H}.\mathbf{x} + \hat{\mathbf{c}}^{\mathbf{T}}.\mathbf{x}, && (10)\\
&g(\mathbf{x}) \geq 0, && (11)\\
&h(\mathbf{x}) = 0 && (12)\\
&\text{Determine,} && (13)\\
&\mathbf{x}^*(\hat{H}, \hat{\mathbf{c}}) = \arg\min_{\mathbf{x}} f(\mathbf{x}, \hat{H}, \hat{\mathbf{c}}), && (14)
\end{aligned}
$$

Where $\mathbf{x}^*$ is a function of $\hat{H}$ and $\hat{\mathbf{c}}$ because it is dependent on the predictions 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 [2].

4.1 Decision Focused learning (DFL)

The core aspect of this approach is to incorporate the *decision* derived from the prediction, $\hat{\mathbf{c}}$, into the learning algorithm. This is done by quantifying the accuracy of the decision *based on* the prediction ($\mathbf{x}^*(\hat{\mathbf{c}})$), which we call *Regret*. The Regret is formally expressed below.

$$
Regret(\mathbf{x}^*(\hat{\mathbf{c}})) = f(\mathbf{x}^*(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^*(\mathbf{c}), \mathbf{c}) \tag{15}
$$

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'. Here, as $\hat{\mathbf{c}} \to \mathbf{c}$, $Regret \to 0$. So, the better the prediction, the lower the regret. Instead of making a point estimation, it is also possible to estimate the distribution of $\hat{\mathbf{c}}$ so that it is possible to guard against the worst case using the distributional knowledge, however more challenging. We refer the reader to [6] for a more comprehensive overview of the learning algorithm.

4.2 Express Symbolically, then optimize (ESO)

In this method we follow a 2-stage procedure. In stage 1 we use symbolic regression to determine the symbolic expression of the function that describes the data. In stage 2 we use this function and the given constraints to determine the optimal combination of inputs. Any symbolic regressor is viable. In the future, we shall experiment with KANs and an ensemble of KANs in a **Mixture of Experts** [7] manner to solve several optimization problems simultaneously.

5 Conclusions

In this project, we discussed the techniques that are used to solve optimization problems and how data driven methods can now adopt those techniques thanks to symbolic regression. The approach is to first use symbolic regression and then optimize. Besides analysing performance of different symbolic regressors, it is important to study techniques to evaluate this 2 stage approach. Since, we are essentially *deriving* decisions from data, we imagine the type of evaluation that is needed is post-hoc evaluation, after the decisions have been prescribed and have been executed and further data has been collected on the results. This is both a challenge and a novel way of evaluating predictive models that affect downstream decisions. It brings into light that the prediction may not be end of the workflow.

References

[1] Kasun Amarasinghe et al. "Explainable machine learning for public policy: Use cases, gaps, and research directions". In: *Data & Policy* 5 (2023), e5. DOI: `10.1017/dap.2023.2`.

[2] Adam N. Elmachtoub and Paul Grigas. "Smart "Predict, then Optimize"". In: *Management Science* 68.1 (2022), pp. 9–26. DOI: `10.1287/mnsc.2020.3922`. eprint: `https://doi.org/10.1287/mnsc.2020.3922`. URL: `https://doi.org/10.1287/mnsc.2020.3922`.

[3] Rudolf Kruse et al. "Introduction to Evolutionary Algorithms". In: *Computational Intelligence: A Methodological Introduction*. Cham: Springer International Publishing, 2022, pp. 225–254. ISBN: 978-3-030-42227-1. DOI: `10.1007/978-3-030-42227-1_11`. URL: `https://doi.org/10.1007/978-3-030-42227-1_11`.

[4] Ziming Liu et al. *KAN: Kolmogorov-Arnold Networks*. 2024. arXiv: `2404.19756 [cs.LG]`. URL: `https://arxiv.org/abs/2404.19756`.

[5] Nour Makke and Sanjay Chawla. "Interpretable scientific discovery with symbolic regression: a review". In: *Artificial Intelligence Review* 57.1 (Jan. 2024), p. 2. ISSN: 1573-7462. DOI: `10.1007/s10462-023-10622-0`. URL: `https://doi.org/10.1007/s10462-023-10622-0`.

[6] Jayanta Mandi et al. “Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities”. In: *Journal of Artificial Intelligence Research* 80 (Aug. 2024), pp. 1623–1701. ISSN: 1076-9757. DOI: 10.1613/jair.1.15320. URL: http://dx.doi.org/10.1613/jair.1.15320.

[7] Saeed Masoudnia and Reza Ebrahimpour. “Mixture of experts: a literature survey”. en. In: *Artificial Intelligence Review* 42.2 (Aug. 2014), pp. 275–293. ISSN: 1573-7462. DOI: 10.1007/s10462-012-9338-y. URL: https://doi.org/10.1007/s10462-012-9338-y (visited on 11/28/2024).

[8] Aditya Mayte et al. “Case Study: Applying Decision Focused Learning in the Real World”. In: 2022.

[9] Ninareh Mehrabi et al. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607.

[10] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0. URL: https://www.nature.com/articles/323533a0.

[11] P. Whittle. “Restless Bandits: Activity Allocation in a Changing World”. In: *Journal of Applied Probability* 25 (1988), pp. 287–298. ISSN: 00219002. URL: http://www.jstor.org/stable/3214163 (visited on 12/12/2024).

[12] Wikipedia contributors. *Test functions for optimization — Wikipedia, The Free Encyclopedia*. [Online; accessed 13-December-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Test_functions_for_optimization&oldid=1262242809.

[13] Runpeng Yu, Weihao Yu, and Xinchao Wang. *KAN or MLP: A Fairer Comparison*. 2024. arXiv: 2407.16674 [cs.LG]. URL: https://arxiv.org/abs/2407.16674.