Complexity Class Analysis with Machine Learning

CS4991 Capstone Report, 2023

Mitchell Hong
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
Mch6eq

ABSTRACT

Although computation power has increased greatly in the last few decades, computer programs remain bottlenecked by inefficient, non-elegant solutions. To address this issue, a machine learning model can be developed to analyze the semantics and logical framework of computational problems. The model would take as input a specifically formatted string that encapsulates the logical structure of a problem. In return, it would output the problems most likely complexity class. With this, computer scientists would be able to identify disconnects from their intuition and the physical complexity lower bounds for the respective problem.

To construct such a model, a large corpus of problems along with their descriptions must be gathered. Once sufficient training data has been gathered, a Natural Language Processing (NLP) model can be employed to decompose the problems to their base logical structure. The model will then identify distinct characteristics associated with each complexity class and subsequently predict the appropriate complexity class to which a given problem belongs. Upon successful development, the next phase of this model would be to also output the reason why the model classified a given problem to a specific complexity class, further allowing computer scientists to make algorithmic breakthroughs.

1. INTRODUCTION

In CS 4102 Algorithms, complexity analysis was the centralized, main focus of the course. Complexity can be measured in two different ways: spatially and temporally. Temporal complexity -- how much time a Turning machine requires -- is the main focus of the model. However, later iterations of the model can account for spatial complexity -- the amount of space required by a Turning Machine -- as well.

The CS 4102 tasked students with developing intuition in determining the complexity class of various problems. For instance, a student may read the following description: *"In computer programming, iterating through a list involves traversing each element in the list one by one, usually employing a loop construct such as a 'for' or 'while' loop."*, and determine that the complexity of the problem described is linear as it involves simply iterating through a list. The student's growing intuition for complexity classes largely stems from repeatedly recognizing unique characteristics specific to each class. With a large amount of clean data, a machine learning model could be constructed to accomplish the same task plus much more. In other words, a well-constructed model could be a powerful tool that revolutionizes the field of problem complexity.

2. RELATED WORKS

The groundbreaking study by Webb et al. (2023) presents a comparison between the performance of GPT-3, a Large Language Model (LLM), and human participants in zero-shot analogical reasoning tasks, challenging the traditional belief that "fluid intelligence" is a uniquely human trait. The study provides empirical evidence that GPT-3 not only matches but in some conditions surpasses human performance in tasks traditionally used to measure "fluid intelligence." These findings offer compelling support for the notion that LLMs can contribute key insights into the complexity class of problems, reinforcing the idea that models like GPT-3 are not merely text generation tools but can significantly enhance our understanding of complex reasoning in both human and computational contexts.

Beame et al. (1998), delves into the complexity hierarchy within NP search problems to better understand their relative computational hardness. The author employs a variety of techniques to establish a framework for comparing the complexity of NP-complete problems. This study offers critical insights into the structural relationships among different NP search problems and highlights the complexity landscape within the class of NP problems. The work stands as a cornerstone in theoretical computer science, illustrating the nuanced relationships that define computational complexity for a broad class of problems.

3. PROPOSED DESIGN

The process of constructing a machine learning model usually boils down to a few core steps: data collection, model selection, and model fine tuning. These core steps further break down into smaller sub steps.

3.1 Data Collection

Arguably the most important step in this process, data collection will be the major challenge to overcome. While there are descriptions of problems and their associated complexity analyses readily available on the internet, a centralized, uniform repository holding this information does not exist. This means that a significant amount of effort and time will go into pooling unstructured data and finding optimal ways to store it. Although the use of a LLM provides slack in the way we represent data, it is still important to standardize the data for the best results. For example, consider these two sentences "*The dog and cat fought*" and *"The fight was between the dog and cat"*. While the LLM will be able interpret both sentences, the input may be tokenized in different ways, causing a varied output. Further, designing the description of problems will require substantial effort, intention, and focus.

Ensuring the purity and objectivity of the data is paramount. When collecting data, it is crucial to focus solely on the essential elements that define the problem to avoid incorporating noise and bias. By strictly adhering to the semantics of a problem and excluding irrelevant information, we can maintain the integrity of the dataset. If overlooked, this could lead to the integration of unintended biases, skewing our results and potentially leading to inaccurate predictions and insights. Therefore, a meticulous and disciplined approach to data collection is indispensable for preserving the reliability and accuracy of our predictive models, enabling us to draw more precise and unbiased conclusions.

3.2 Model Selection

Model Selection forms the backbone of building a robust machine learning model. It refers to the process of choosing the most suitable algorithm or model architecture that can best capture underlying patterns in the data. This step is crucial because the choice of model significantly influences the performance and reliability of the system in making predictions. It often involves a thorough exploration of various models, ranging from simpler linear models to more complex ones like neural networks, to ascertain which one aligns best with the problem at hand and the nature of the collected data.

In this step, it is pivotal to consider the characteristics and constraints of the problem domain, and develop a firm understanding of the nature of the data. Additionally, it's vital to perform initial assessments through techniques like cross-validation to evaluate the models' performance based on predefined metrics such as accuracy, precision, recall, F1 score, or Mean Squared Error.

For the proposed complexity class model, the proposed workflow includes several steps: initially, one must leverage a LLM, such as GPT-4, to process and tokenize inputs. After normalization of these inputs, they are subsequently run through a classifier model to discern the respective complexity class to which a problem may belong. Additionally, a clustering model can be utilized to identify patterns among the problems. However, employing such a model would necessitate further external analysis to comprehend the rationale behind the grouping of problems and to interpret the observed patterns.

3.3 Model Fine Tuning

Model fine-tuning is a meticulous and iterative process in which hyperparameters and features are adjusted to optimize performance. It is hypothesized that, for this proposed project, substantial effort will be devoted to refining the training data format. This is because identifying the most effective input string to define a problem is likely to involve a degree of experimentation and iterative refinement. Once a solid format is crafted, focus can be placed on feeding the model additional training data. Overall, the model's performance can continuously be improved through the use of these techniques.

4. ANTICIPATED RESULTS

The end goal of this endeavor is to develop a multi-class classifier capable of discerning the complexity class to which a problem pertains. For problems with pre-established complexity classes, the model's predictions should align accurately with these known classifications. Achieving this level of accuracy would allow the model to analyze problems with undetermined lower-bound complexities reliably. The primary objective is to explore instances in which the intuitions of computer scientists diverge from the model's predictions. This way, the solution to a problem can be analyzed from a completely different perspective, allowing for new, innovative solutions to be explored. Additionally, the model could employ clustering techniques to group problems based on similarities, offering scientists a structured foundation for drawing parallels and distinctions between problems. This would potentially illuminate new pathways for exploration and understanding in complexity class analysis. While the model's autonomous utility may be limited, its value as an analytical tool in the realm of complexity class analysis is substantial, paving the way for groundbreaking insights and discoveries.

5. CONCLUSION

The realm of complexity analysis has remained pivotal in computational theory, with its implications impacting the efficiency and effectiveness of computer programs and algorithms. This research endeavors to bridge the intricate world of complexity classes with the capabilities of machine learning, specifically through the utilization of Natural Language Processing models. By harnessing the power of large datasets and advanced machine learning techniques, the proposed model aims to autonomously discern the complexity class of computational problems, thus offering a new lens through which computer scientists can view and analyze these challenges. The significance of this project lies not only in its potential to accurately classify complexity, but also in its capacity to challenge and refine human intuition on the subject, catalyzing a revolution in how we approach and understand computational problems. With the successful implementation and further refinement of this model, the landscape of complexity class analysis stands to be transformed, ushering in a new era of enhanced understanding and breakthroughs in the domain of computer science.

6. FUTURE WORK

A substantial enhancement involves not just the model's ability to classify computational problems into their respective complexity classes, but also its capability to elaborate on its predictions. Future iterations of the model will be equipped to output the specific anomalies or irregularities it discerns, providing a comprehensive explanation for designating a complexity class that diverges from the pre-existing or expected label. Such a feature would be invaluable for computer scientists, reducing the analytical overhead traditionally required to decode the intricacies of a problem's complexity. Instead of spending considerable effort delving deep into the problem to understand its complexity nuances, scientists could simply review the detailed rationale provided by the model. This approach not only streamlines the analytical process but also enriches the insights learned, potentially paving the way for more intuitive and efficient solutions to complex computational challenges.

7. ACKNOWLEDGMENTS

I would like to extend my gratitude to Professor Rosanne Vrugtman for her invaluable insights and thorough review of this paper. Her comprehensive feedback greatly enhanced the quality and depth of this research. I am also extremely thankful for the dedicated staff of UVA for teaching a curriculum that is not only enriching but also inspiring. Their commitment to excellence and passion for teaching has been instrumental in shaping this idea.

REFERENCES

Webb, T., Holyoak, K. J., & Lu, H. (2023). Emergent analogical reasoning in large language models. *Nature Human Behaviour*. https://doi.org/10.1038/s41562-023-01659-w

Beame, P., Cook, S., Edmonds, J., Impagliazzo, R., & Pitassi, T. (1998). The relative complexity of NP search problems. *Journal of Computer and System Sciences*, *57*(1), 3–19. https://doi.org/10.1006/jcss.1998.1575