CS181P Final Project Proposal #2

Santi Santichaivekin, Jingyi Liu, Jacob Boerma, Lorraine Zhao, Princewill Okoroafor
Harvey Mudd College and Claremont McKenna College
Claremont, California, USA

1 Team

The team includes Santi Santichaivekin, Jingyi Liu, Jacob Boerma, Lorraine Zhao, and Princewill Okoroafor. The team name is Go Pika Pika! The team logo is provided below.

2 Project

We will attempt the second project listed on the project ideas:

> ML, Search, and Compression. It has been shown that ML is a form of search (we will see this during the course, for example in Montanẽz (Chapter 3)), and also shown that ML can be viewed as a form of compression (as in minimum description length approaches). Can we show compression is a form of search, search a form of compression, search a form of ML, or compression a form of ML, to make a triangle of two-way equivalences? Or taking a subset of these equivalences, can we apply a set of results from one domain to the other, such as forming no free lunch theorems for compression?

We hypothesize that all the three problems can be phrased as one another. If we prove the three-way equivalency among the three problems, then we can apply different theories from one field to another. Also, this problem is applicable to the real world, where we can solve actual real world problems by directing them as either ML, search, or compression.

3 Initial Plan of Attack

We will start with understanding the concepts of ML, data compression, and search individually, and then trying to prove the "No Free Lunch" theorem for data compression among other theorems of search that might be applied to data compression. We can then look for algorithms for data compression and obtain insights about how it might be framed as a machine learning problem and search.

4 Separation of Work

For now, we will each read the papers and get on the same level of understanding. Once we have a good understanding of ML, compression, and search, we might split up work based on sub-problems that appear. We have divided the readings to different team members, so that one person can focus on one or two readings, and then update the rest of the team members on what he/she learned.

- Jacob will read about No Free Lunch on a high level, and will try to understand the Famine of Forte paper and how we might apply the same concepts to compression schemes.
- Santi will be trying to understand the No Free Lunch Theory and apply it to compression.
- Lorraine will go over her part of the readings and get a better understanding of Machine Learning and Searches, and try to link the two by establishing an equivalent relationship.
- Rose will be reading about data compression algorithms and see how to generalize them as search problems.
- Princewill will also be reading about data compression algorithms and how they can be generalized as search problems.

5 Literature Review

We have found some papers and textbooks that will be useful for our research. We will read the suggested readings for the next few lectures in class, as well as the following papers/articles.

Why ML Works `http://www.cs.cmu.edu/~gmontane/montanez_dissertation.pdf` [Lorraine]

"Famine of Forte" `https://arxiv.org/pdf/1609.08913.pdf` [Jacob]

"Probabilistic machine learning and artificial intelligence" `https://www.nature.com/articles/nature14541` [Rose]

In his paper "Probabilistic machine learning and artificial intelligence", Ghahramani provides an overview of the probabilistic approach to machine learning and the state-of-art advances in this field. Specifically, he explores five areas that fall under this framework including probabilistic programming, Bayesian optimization, probabilistic data compression, automatic model discovery, and hierarchical modeling. Under the framework of probabilistic modeling, Ghahramani views machine learning as a process of inferring the most plausible model from a given data set, which is in turn used to improve performance on future predictions. He points out the advantages of using a probabilistic framework by arguing the central role of uncertainty in machine learning models and identifies three sources of uncertainty that are typically present in modeling (452-453). The uncertainties consist of the measurement noise of the data set, the unknown parameters in a given model, and uncertainties about the general structure of the mode. Thus probabilistic modeling comes as a natural remedy for dealing with various sources of uncertainties and identifies the best model in accordance with data. Aside from its ability to represent uncertainties in the model, probabilistic modeling

also brings the advantage of flexibility through the use of Bayesian non-parametric models, which grow in complexity as the data set grows (454). Such flexibility allows machine learning models to capture the regularities of the data and have better performance. Interestingly, he points out that Bayesian nonparametrics is equivalent to a neural network with infinite hidden units. In the section about Bayesian optimization, he formulates the optimization process as a sequential decision problem—a search problem in our words—where you use a probabilistic distribution of the function values and determine the next query by choosing the area with the most uncertainty (and thus the most information gain once the value is reviewed) (456). Here we notice a connection between a specific type of machine learning problems and search is made through the bridge of the probabilistic framework. In the section about data compression, he claims the equivalence of data compression and probabilistic modeling as "two sides of the same coin" and highlights the important role that Bayesian machine-learning models play in the field of data compression (456). The process of optimizing data compression can be viewed as a process of extracting the best model from data, and because of the flexibility provided by Bayesian non-parametric models, some of the best data compression algorithms employs the same idea to achieve better compression rate. Here, a connection is made between machine learning and data compression through the same bridge of probabilistic modeling. Thus from the review of this literature, it is evident that the probabilistic framework can be profound in trying to equate the concepts of machine learning, compression, and search, and is thus an area of interest that we can explore in more details in our project.

"Generalization as Search" `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.5764&rep=rep1&type=pdf` [Lorraine]

"Data Compression Explained" `http://mattmahoney.net/dc/dce.html#Section_13` [Rose] [Princewill]

Data Compression Explained is an online book by Matt Mahoney that tries to explain how data compression works. It starts with an information theoretic introduction to data compression and goes on to discuss different codes, models, transforms and lossy compression of images, audio and video. In the information theoretic introduction which we care more about, Mahoney highlights that information theory places hard limits on what can and cannot be compressed losslessly, and by how much:

1. There is no such thing as a "universal" compression algorithm that is guaranteed to compress any input, or even any input above a certain size. In particular, it is not possible to compress random data or compress recursively.

2. Given a model (probability distribution) of your input data, the best you can do is code symbols with probability p using $\log_2 \frac{1}{p}$ bits. Efficient and optimal codes are known.

3. Data has a universal but uncomputable probability distribution. Specifically, any string x has probability (about) $2^{-|M|}$ where M is the shortest possible description of x, and $|M|$ is the length of M in bits, almost independent of the language in which M is written. However there is no general procedure for finding M or even estimating $|M|$ in any language. There is no algorithm that tests for randomness or tells you whether a string can be compressed any further.

Mahoney also explains that modeling is not computable citing work by Solomonoff (1960, 1964), Kolmogorov (1965), and Chaitin (1966) who independently proposed a universal a-priori probability distribution over strings based on their minimum description length. The algorithmic probability of a string x is defined as the fraction of random programs in some language L that output x, where each program M is weighted by $2^{-|M|}$ and $|M|$ is the length of M in bits. This probability is dominated by the shortest such program. We call this length the Kolmogorov complexity $KL(x)$ of x. The best compression we can achieve for any string x is to encode it as the shortest program M in some language L that outputs x.

Mahoney also discusses how compression is an artificial intelligence problem by explaining how prediction or compression could be used to test or measure understanding.

"Minimum Description Length" https://www.cs.cmu.edu/~aarti/Class/10704/lec13-MDL.pdf [Princewill]

The minimum description length (MDL) criteria in machine learning says that the best description of the data is given by the model which compresses it the best. This is because learning a model for the data or predicting it is about capturing the regularities in the data and any regularity in the data can be used to compress it. Thus, the more we can compress a data, the more we have learnt about it and the better we can predict it. The ideal version of MDL, given by the Kolmogorov Complexity, which is defined as the length of the shortest computer program that prints the sequence of observed data and halts, is uncomputable.

Practical versions of MDL are either based on one stage universal codes (known as refined MDL) or two-stage codes (known as crude MDL). The refined MDL suggest that, for a given class of models, pick the model which minimizes the worst case redundancy on the data. This leads to precisely the universal models such as mixture models and normalized maximum likelihood (NML) model. The crude MDL or two-stage code approach is based on the notion that we can specify the descriptive properties of a model for data in two stages - (i) encode the model with some codelength $L(q)$, (ii) encode the data using the model with codelength $L_q(x^n)$. Now pick the model which minimizes the total codelength of the two-stage code: $q_\gamma(x^n) = \arg\min_{q \in Q_\gamma}\{L(q) + Lq(x^n)\}$

The MDL principle can also be used for classification or regression. The two-stage MDL procedure for selecting the best model in a given class would be the regularized least squares framework.

No Free Lunch For Optimization https://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf [Santi] [Jacob]

The No Free Lunch Theorems for Optimization paper establishes that under an oracle model of search, two algorithms that use the same number of oracle reveals have the same expected performance if all cost functions have the same likelihood of being chosen. Wolpert and Macready proves this by showing that any algorithm reveals the same sequence of cost on average regardless of what has been previously revealed to the algorithm. The formal proof is done by induction using a statistical model, but it could have been done without statistical model too. The paper also goes on to prove the NFL result for time-dependent cost functions and relate the result to other topics such as information theory and geometry. I have not looked into these topics yet as it seems to be more complex.

The search model used in the paper includes simulated annealing and evolutionary algorithm (where each species is independently evaluated). It does not include techniques like branch and bound or coevolutionary algorithm (where species are evaluated against each other). The authors also went on to prove that there is free lunch for coevolutionary algorithms, that is, some algorithm performs objectively better compared to others.

The authors also mention that they had proved a similar result for statistical inference (machine learning) which I have not explored. I think I will need to spend some time to understand the proof for optimization better before delving into the other one.