Maximum Entropy Inference for Machine Learning

Peter Cheeseman and Nick Candau

July 11, 2023

Abstract

In this paper, we argue that machine learning can be cast as a search through model space to find models that best describe the data. Here, "best" means maximally compresses the data. A model compresses the data if it value differs significantly from its corresponding MaxEnt expectation. Any such significant models (constraints) reduce the entropy of the data. We give a procedure for performing this open-ended incremental search in model space. Data compression is formalized by a method known as Minimum Message Length, and this in turn is a discretized version of Bayesian inference using MaxEnt priors and likelihood functions. The MaxEnt method described here is designed to automate the hypothesize and test cycle, allowing machine learning to operate autonomously. The explicit/declarative representation used in our MaxEnt method allows learned knowledge to accumulate in a form that makes it easy to reason about and check for consistency. It is also a useful source of prior knowledge to help guide the search for further knowledge. This declarative representation allows new model hypotheses to be generated beyond those traditionally used in machine learning.

1 Introduction

For over two millennia, human beings have sought a systematic way to explain the world around them. One approach, based on empiricism, has yielded such impressive results it has become inextricably interwoven into modern life. This is the hypothesize and test cycle (Fig. 1), also known as the scientific method. While intellectual giants such as Newton, Galileo, and Bacon are given much of the credit for developing and formalizing this cycle, its roots are much deeper, as exemplified by thinkers such as Aristotle, and informally much further back as a general method for understanding the world. Until recently, humans were responsible for every step of this cycle. This changed with the widespread availability of computers in the late 20th century which allowed some of the steps to be automated. To achieve Artificial General Intelligence (AGI), this cycle must be fully automated. In this paper, we propose using logic and probability as a general language to represent all hypotheses, and Maximum Entropy Inference (MEI) as the universal inductive inference method as components of this automation. Since only part of the hypothesis and test cycle has been automated to date, current Machine Learning (ML) should more accurately be referred to as machine *assisted* learning. Recently, ML has widely come to mean using Deep Neural Nets (DNNs) for performing ML. In contrast, we use ML to refer to the earlier idea of using computers to do some of the hypothesizing and testing autonomously, such as in statistical data analysis, as well as in DNNs.

The recent advent of Large Language Models (LLMs), such as OpenAI's ChatGPT, represents a paradigm shift in ML. While still in its infancy, the remarkable successes they've had in generating human-like text and solving novel problems testifies to their potential to automate more of the steps of the hypothesis and test cycle. Despite their successes, however, LLMs have limitations including the opacity of the model (lack of explainability) and occasionally generating incorrect responses (hallucinations). The architecture of LLMs is so different from traditional ML that their success challenges many of the assumptions and ideas underlying traditional ML. Currently, no LLM has fully automated the hypothesize and test cycle. This is partly by design as LLMs are set up to answer user queries, not generate their own. Additionally, their limitations may be more fundamental in that they don't have a declarative representation of the concepts embodied in the parameters of the neural nets. This lack of explainability poses a particular challenge for hypothesis generation based on previous results [9].

The hypothesis and test cycle begins with a question or problem the agent wants to solve, then data is collected that the agent judges to be relevant the question. The rest of the cycle can be thought

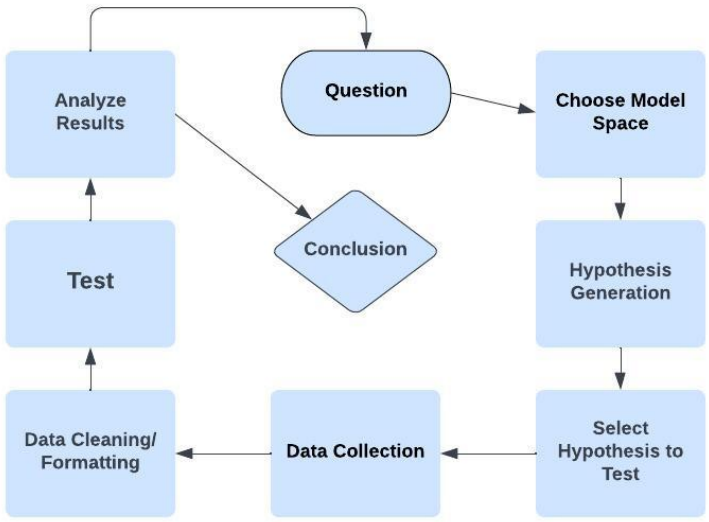

Figure 1: The Hypothesis and Test Cycle

of as a search through model space to find the model(s) that best explains the data. After choosing a model space based on prior domain knowledge, a specific hypothesis (model) is selected to be tested within that space. Based on this test, the new model is either confirmed or rejected. In either case, the cycle continues as long as desired by the researcher.

2 An Example

In order to explain and make concrete the theory developed in this paper we use the same ML example throughout. Specifically, we apply our method to English text data to see how much known English structure can be found using purely automated ML methods. Natural language text is the same type of data used in training LLMs. This example problem is a special case of the more general problem of discovering model(s) in data consisting of a sequence of discrete symbols drawn from a fixed set of possibilities. Such data include DNA sequences, musical scores, stock market data, etc., as well as text. No prior language knowledge is assumed here; the only information given to the learning system is that the data is in the form of a *sequence* of symbols drawn from a given finite alphabet. The ML approach described here can be applied to any of the types of sequence data mentioned above. The text data can be represented as specific logical statements such as: At[Republic, 1] = "T", At[Republic, 2] = "h", etc., where "Republic" refers to an English translation of Plato's republic, and the numbers refer to the index positions in the text.

We know that the model space for our English text example is hierarchical, so the model space should include the possibility of hierarchical models. The levels of this hierarchy for this problem are, starting with the lowest level, firstly finding classes of characters (letters, numbers, punctuation,..), then words. The next level up is word morphology (tense and number agreement, prefixes and suffixes), then grammar (sentences, phrases, etc.), then document structure (paragraphs, sections, chapters, etc.), and finally the semantic level where the meaning of the text is uncovered. Typically we are only interested in the semantics of the text, but it is necessary to go through all the lower levels of the modeling hierarchy to get to the semantic level. We have only succeeded in discovering the lowest levels of this hierarchy, but expect that the methods used will allow the higher levels of the hierarchy to also be discoverable. Since both children and LLMs can learn up to the semantic level, it is reasonable to expect that fully automated ML can do the same.

3 Maximum Entropy Inference (MEI)

Historically there have been two fundamentally different approaches to inductive inference used in the "test" box of the hypothesize and test cycle (Fig. 1)–they are Bayesian Inference and Maximum Entropy Inference (MEI). This paper is based exclusively on MEI, as developed below. We start by presenting the Principle of Maximum Entropy (PME) on which MEI is based.

3.1 The Principle of Maximum Entropy (PME)

The basic idea of PME is to choose a particular probability distribution from the set of possible probability distributions over the joint model space, given a set of constraints that apply to this joint space. The chosen probability distribution is intended to represent the agent's "best" understanding of that domain given the set of constraints. The idea of "best" used here is to choose the distribution that assumes the least information given the constraints. PME assumes that the known constraints are the only significant constraints in the domain–we call this assumption the *constraint completeness assumption*. If there are reasons to doubt this assumption, then the accuracy of the PME probabilities as a representation of the agent's knowledge of the domain is in doubt.

The representation of probabilities in the MaxEnt formalism used here is in the form of a discrete joint probability distribution that assigns a probability to every possible instance that conforms to the given general data format. In our text example, the joint probability distribution of all possible character sequences can be represented by: $P^{j,k,l,\cdots}_{1,2,3,\ldots}$, where the subscripts are the positions in the sequence, and the superscripts are indices into the fixed character set at the corresponding position. When this probability joint space is fully constrained, there is only one possible character sequence, and this is the given text data. In contrast, at the beginning of the MEI cycle when no constraints have been tested, MaxEnt assigns uniform probability to every possible character sequence–an extremely large set of possibilities. As the MEI cycle continues, some possible sequences are excluded (logical constraints) while some possible sequences are made more or less probable as a result of adding constraints. We note that if the set of logical constraints is sufficient to fully constrain the possibilities to a single sequence, then this is equivalent to entailment in logical inference. Generally, the significant constraints found by the MEI method do not uniquely constrain the set of possible sequences, so further specific information is necessary to reduce the remaining uncertainty to zero. Without the specific information, the remaining uncertainty can be thought of as the residual "noise"–i.e. the component of the data that is not predictable and so must be specified explicitly.

Using the above representation of the probabilities of the joint space, the total entropy of a given probability distribution in this space is defined by:

$$H = -\Sigma P^{j,k,l,\cdots}_{1,2,3,\ldots} Log[P^{j,k,l,\cdots}_{1,2,3,\ldots}], \tag{1}$$

where the sum is over all possible combinations of subscripts and superscripts. The entropy H in formula (1) is the "total" entropy. PME requires finding the joint probability distribution that maximizes H subject to whatever constraints are being enforced. If the constraints are sufficient to constrain the probability distribution to a single possibility, then $H = 0$, otherwise, $H > 0$. The general solution to the constrained PME problem can be found using the method of Lagrange multipliers, as shown in [1, 5], and is given by:

$$P^{j,k,l,\cdots}_{1,2,3,\ldots} = e^{-\lambda_0 - \lambda_j - \lambda_k - \cdots}, \tag{2}$$

where every λ parameter corresponds to a particular constraint. For example, the λ_0 parameter corresponds to the normalization constraint for the joint probability distribution. λ_j, for example, corresponds to the constraint that the jth character has a probability of P_j independent of position in the text sequence.

Once the full joint ME probability distribution is found using equation (2), *any* marginal joint probability can be found by summing over the indices not included in the marginal definition. Any desired conditional probability can be found by taking the ratio of the corresponding marginal probabilities. Because there are as many λ parameters as there are given constraints, the λs can be determined from the given constraint values. This determination can be performed by solving a set of (nonlinear) equations, where these equations are formed by setting the corresponding marginal probability equal to the empirical constraint value. Since any marginal probability computed using equation (2) is a function of all the λ parameters not included in the summation, this means that each constraint equation is a

function of a set of λs, and so this set of equations must be solved using nonlinear equation solving methods to find the values of all the λs. When these λs are determined, the full joint ME probability given by (2) is fully determined. For further details of this method for solving equation (2) see [1].

As an example of using equation (2), consider the case where the only constraints are the empirical probabilities for the individual characters independent of position in the text. In this case, equation (2) reduces to:

$$P_{1,2,3,\cdots}^{j,k,l,\cdots} = P_1^j P_2^k P_3^l \cdots \tag{3}$$

That is, the joint probability for any choice of the indices $j, k, l, \cdots$ is given by a product of the individual character probabilities at each position in the text. Another way of saying the same thing is that under this assumption, the PME gives the marginal independence result one expects. The PME gives similar conditional independence formulas for some cases of overlapping constraints. For constraint sets where equation (2) cannot be factored, there is no closed-form formula for the MaxEnt distribution, but the MaxEnt distribution can still be computed numerically. It is these independence results that lead us to claim that the PME is a generalized principle of independence. That is, the PME yields the standard independence formulas in simple cases but can be thought of as giving a maximally independent probability distribution even in cases where there is no simple formula.

An obvious question is why should MaxEnt be used as a universal prior. There are three justifications for MaxEnt in the literature that we are aware of–they are:

1. The least information principle. This principle says that the MaxEnt choice is the one that assumes the minimum information that includes the information contained in the constraints used in the MaxEnt calculation. Put another way–any other choice would be assuming information that the user does not have.

2. Argument from statistical mechanics. This argument assumes that any closed physical system is either in a state of maximum entropy or evolving in that direction. This means that assuming the physical world from which the data is drawn is in a state of maximum entropy is a good prior assumption. Any significant deviation from a state of MaxEnt (subject to current constraints) requires evidence from the data before being accepted. Jaynes in [4] showed that the MaxEnt approach used in statistical mechanics can be applied more generally to data that is not the result of measurements of physical systems, such as our text example.

3. ME (subject to the known constraints) is a generalized principle of independence (see above), and independence is the norm in the real world.

These justifications essentially express the same idea in different forms–i.e., they are equivalent, although this is not obvious.

3.2 The Maximum Entropy Inference (MEI) Method

The MEI procedure shown in Fig. 2 is a formalization and generalization of the procedure given by example in [5]. Our expanded version of MEI has four nested loops:

1. modified data loop (hierarchical models)

2. which model space to use next;

3. what particular constraint within that model space to test next;

4. whether the chosen constraint is significant given its expected ME value.

In the MEI procedure shown in Fig. 2, any significant constraints are added to the accumulated constraint store so that they can be used in computing ME values for future significance testing. The accumulated constraints can also be used to modify the data. This modified data can then be used in the outer loop in Fig. 2, creating a form of hierarchical learning. MEI assumes that the current set of constraints is a complete model of any structure in the data, but tests this assumption by evaluating each new proposed constraint against the corresponding current MaxEnt value.

The MEI cycle continues until the model space generator runs out of models to test, or runs out of time. The inner loop for testing constraints stops when there are no more specific constraints to test.

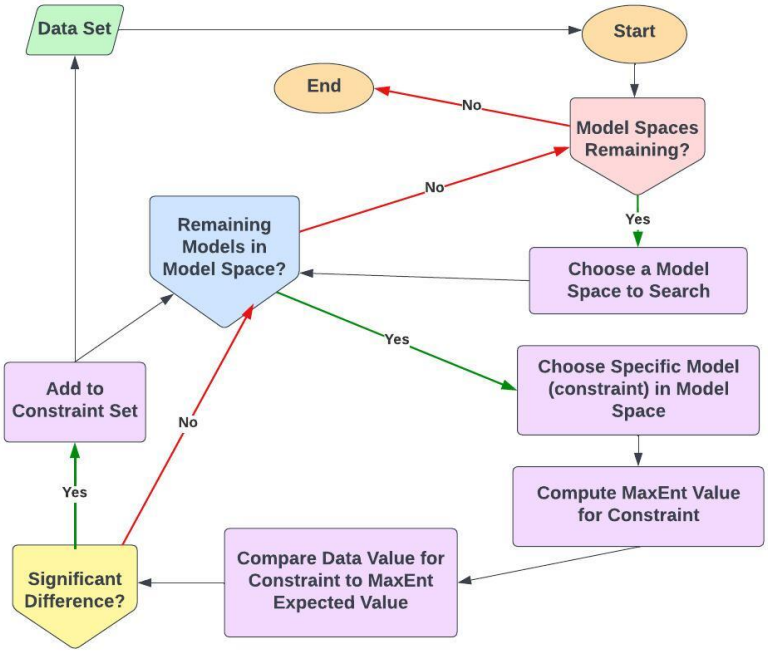

Figure 2: The MEI Flowchart

Note that in Fig. 2 the choice of model space is followed by an exhaustive search of constraints in that space before choosing another model space. Future MEI procedures could modify this MEI procedure to allow flexible interleaving of these two steps. Although the procedure shown in Fig. 2 is serial, it could be easily parallelized–in particular, alternative model spaces could be explored in parallel, and the constraint testing within each model space could also be parallelized. However, hierarchical model discovery can only proceed serially since each level builds on the previous levels.

In our text example, the initial data is the text to be analyzed. An example model space during MEI search could be text segmentation models, while a specific candidate segment is the hypothesis/constraint to be tested. If the probability of this segment is found to differ significantly relative to the expected MaxEnt value for the unsegmented text, the data is then split between this segment and the rest. MEI now proceeds separately on these two data sets. Although these split learning problems proceed independently, they can share information through priors relating models in the two spaces and thus reducing the information required to specify the separate models–a form of transfer learning. Other learned models, such as words or phrases, can be added to the data so they can be used as building blocks for higher-level models, such as sentences, and so on.

3.3 Significance Testing in MEI

The innermost step in the MEI method is to decide whether the current constraint is significant or not relative to the current MaxEnt expectation. To perform this significance test we use relative Bayesian hypothesis testing. If H_1 is the hypothesis (constraint) to be tested, H_0 is the hypothesis that the data value of this constraint is sufficiently close to the MaxEnt value (calculated using all the currently known significant constraints), D is the data and B is the background information (a constant in all terms), then by Bayes theorem the relative posterior probability ratio is given by:

$$\frac{p(H_1|D,B)}{p(H_0|D,B)} = \frac{p(H_1|B)}{p(H_0|B)} \frac{p(D|H_1,B)}{p(D|H_0,B)}. \tag{4}$$

If the right-hand side of (4) is > 1 then $p(H_1|D,B) > p(H_0)|D,B)$, and in this case we judge the constraint to be significant, otherwise not. The first ratio on the right-hand side of equation (4) is the prior probability ratio of the two hypotheses H_1 and H_0. If we assume equal prior probability for these

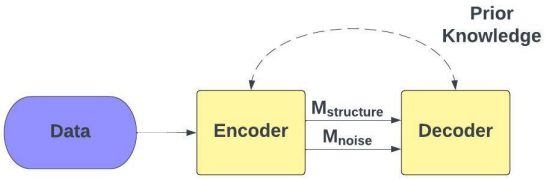

Figure 3: Data Communication

two hypotheses, then this term is 1, otherwise, we can choose the prior probability ratio to represent any prior knowledge we may have. The second ratio on the right-hand side of (4) is the likelihood ratio, where the denominator $p(D|H_0, B)$ is the probability of the data given the MaxEnt assumption (under the current constraints), and the numerator is the probability of the data given the assumption that the data value is not given by MaxEnt.

For example, in evaluating the MaxEnt probability of the bi-gram "th" when the 1st order character probabilities (constraints) are known, the MaxEnt probability is given by: $p(\text{“}th''|D, B) = p(\text{“}t''|D, B)p(\text{“}h''|D, B) = p_{ME}$–i.e., MaxEnt assumes probabilistic independence in this case. Using this MaxEnt probability of "th", we can calculate the likelihood of getting a count of n instances of "th" in the data using the Binomial distribution: $p(\text{“}th''|D, B) = Binomial[n|N, p_{ME}]$, where N is the total length of the text, and p_{ME} is the MaxEnt probability of the pair. That is, the binomial probability gives the probability of counting exactly n occurrences of "th" in the text using the MaxEnt probability. To find the probability $p(D|H_1, B)$ we have to determine the probability of getting exactly n counts with no information about n except that $n \geq 0$ and $n_{max} \leq min[n_t, n_h]$. If we assume a uniform (MaxEnt) probability for each of these possible n values, then $p(D|H_1, B) = 1/(n_{max} + 1)$. With all the terms in (4) now specified, the relative significance of H_1 to H_0 can be decided.

3.4 Information Theory and Minimum Message Length (MML) Induction

We now present the basics of Shannon's information theory [7] needed for significance testing, along with its Minimum Message Length (MML) equivalent and relate these theories to the significance test we use in MEI. In information theory, a new quantity called "Information" is defined by the formula $I_j = -Log[P_j]$, where P_j is the probability that the jth event will occur. Essentially, I_j is a measure of surprise experienced by an agent who is informed that the jth event occurred, where the prior expected probability of the event is P_j. If $P_j = 1$ then $I_j = 0$, i.e. the agent experienced no surprise on learning that an inevitable event actually occurred. Likewise, the agent experiences a lot of surprise if a very unlikely event occurs. If Log is in base 2, the resultant information is measured in bits. The essential result of information theory is that the theoretical minimum number of bits needed for one agent to communicate data (events that occurred) to another agent over a perfect channel (as in Fig. 3) is given by the total information $H_{tot} = \Sigma_j I_j$ needed to specify which events occurred, and H_{tot} is the total entropy of the data. Likewise, the average entropy H_{av} is defined to be the average information per symbol in the data and is defined by $H_{av} = \Sigma_j P_j I_j$. Following Jaynes [5], we identify Shannon's H_{tot} with the entropy H used in MaxEnt calculations above.

The basic setup for information theory is shown in Fig 3. In this setup, an encoding agent has data and wishes to communicate this data to a decoding agent using the minimum number of bits. To do this communication, the encoder searches for the maximal amount of structure in the data, then communicates this structure to the decoder with a message M_{struct}. However, unless the data is generated by a deterministic process, M_{struct} typically does not capture all the information in the data. The resulting residual information has to be encoded in a second message M_{noise} which is also sent to the decoder. When both these messages are sent, the decoder has all the information needed to reconstruct the original data exactly (lossless data transmission). In MEI, the goal of the encoding agent is to find the minimum *total* message length required to communicate the data exactly, where $Length[M_{total}] = Length[M_{struct}] + Length[M_{noise}]$. In other words, the encoding agent must search through a space of possible models of the data to find the particular model(s) that minimizes the length of the total message, M_{total}. In order for communication to take place, both the encoding and decoding agents must have set up prior communication protocols representing their shared prior

knowledge.

Minimizing this 2-part message length as the basis for inductive inference is known in the literature as Minimum Message Length (MML) [3]. If the data is random, no data compression can occur, so the shortest message is the one that encodes data directly–i.e. $Length[M_{total}] = Length[M_{noise}]$. If the data is compressible, then there must be something going on in the domain that is producing such non-random data. This non-randomness is represented in MEI by a set of constraints over the set of possible worlds defined by the prior information shared between the encoder and decoder. Such constraints are how all domain structure is represented in MEI. Note that any discovered significant constraint(s) could be a statistical shadow(s) of an even stronger domain structures that have yet to be discovered. Note also that the observed structure may be the result of some bias in the data collection process, and so is not necessarily indicative of structure in the domain from which the data was drawn. Only if a fully deterministic model of the data is found can the MEI search terminate with the knowledge that there is no further structure to be discovered, otherwise, the search for models that further compress the data could in principle continue.

In MEI, MML is used to pick out the model that maximally compresses the total message. However, there are considerable theoretical and experimental results showing that only picking the "best" model is not optimal for maximizing predictive accuracy. Finding a set of models of approximately equal MML, and using a (weighted) combination of the predictions of these models has been shown empirically to give better predictions on average than just selecting the MML model. This model averaging was given a Bayesian foundation in Solomonoff induction ([8]), and is used extensively in AI ML algorithms, such as in decision trees [2]. The MEI method as described above can easily be modified to find a *set* of low total message length models for use in ensemble prediction rather than just the minimum message length model.

As discussed in [3], MML is a discretized form of Bayes theorem. In MEI, M_{struct} is the information needed to specify all the significant constraints found to date, and M_{noise} is the information required to specify the data given those constraints. The minimum number of bits required to specify all the significant constraints, M_{struct}, is the sum of Log (base 2) of the prior probabilities of each significant constraint. Likewise, the minimum number of bits required to specify the residual data, M_{noise}, is the Log (base 2) of the likelihood function given all the significant constraints. Thus, PME; Bayes theorem using entropic priors and likelihoods; information theory; and MML all have the same theoretical base. MEI ties these theories into an operational form we use as the basis for ML.

4 Model Spaces for MEI

ML can be regarded as a search through model space looking for the model(s) that maximally compress the data. Which model(s) are found by ML then depends on what model space(s) are used in this ML search. The MEI approach to ML assumes that all models can be represented as "constraints", but this leaves open how constraints are represented, and how general this representation is.

4.1 Representation of Models

We assume that initially, no prior knowledge is available other than the data and the general format of the data. So all constraints must ultimately be defined ultimately in terms of this format. In our text data example, the logical data representation is a set of ground axioms such as: At[Republic, 1] = "T", At[Republic, 2] = "h", $\cdots$, At[Republic, 1,213,625] = "D". This set of axioms spells out the entire text of Plato's "Republic" and can be regarded as a logical encoding of text data. The data *format* can be represented by the general axiom: $\forall i\ (i \geq 1)$ and $(i \leq 1,213,625) \Rightarrow$ At[Republic,i] $= C_j$, where i is an index variable ranging over the characters in the text, and j is an index variable ranging over the possible values the characters could take from the set C (C_j). Note that the integer variable i is ordered, because text is ordered, but the j variable is unordered because the possible characters form a *set*, not a sequence. Since this is the only information available to the ML program, all models must be constructed using this format and the specific data alone.

In MEI, constraints can be logical or probabilistic. Logical constraints exclude otherwise possible worlds (microstates), but probabilistic constraints make some possible worlds more or less likely, without excluding any possibilities. Because logical inference is simpler and computationally easier than probabilistic inference, we prefer using logical constraints. An example of a logical constraint is:

occurrences[Republic,"the"] = 16,721–i.e. there are exactly 16,721 occurrences of the string "the" in the Republic text. This constraint excludes any member of the initial set of possibilities that doesn't have exactly 16,721 occurrences of "the" somewhere in the text.

4.2 Hierarchical Models

Hierarchical models are built "bottom-up" from data, starting with the initial data. As new structures/constraints are discovered they can be fed back to become additional data for further learning. This hierarchical model building is shown in Fig. 2 as the outer loop. Another way of saying the same thing is that new models can be built using earlier models as building blocks. For example, sentences are built from words and topics can be built from sentences. Scientific model/theory discovery seems to proceed in this hierarchical fashion.

Another form of hierarchical model learning is for the computer to automatically collect data on which model spaces tend to be successful in capturing structure and which ones don't (meta-data). The success probabilities can depend on the type of data. This metadata can be used to bias searches through model space(s). The metadata can be analyzed using the same MI method as any other form of ML data. Meta-learning is sometimes referred to as "learning to learn".

4.3 Joint versus Serial Model Representation

In LLMs, the learned models are used to predict the next "token", (typically words), and the learning method tries to maximize the predictive accuracy of the next token. This is an example of a serial model. Likewise, Markov models are also serial predictors. However, rather than trying to predict the next word in a sentence serially, it is more convenient to model the word sequence as a parse tree with the words as the terminal nodes of this tree. A parse tree is an example of a joint model representation, where words can relate to each other via the tree, rather than trying to force the information contained in the tree into a serial word predictor. In other words, using MEI to do a search through a space of parse trees (a model space), and compare the MML of alternative parse tree hypotheses, is simpler than trying to force the same information into a serial predictor. It is an open question whether it is always possible to convert a joint model into a serial model or visa versa. A similar issue arises in learning Bayes Nets [6] in the choice as to whether to use directed graphs (conditional probabilities) or undirected graphs (joint probabilities).

4.4 Comparison with other Modeling Languages

The MEI cycle described above requires "explicitly" deciding on which models space to use and which particular hypotheses (constraints) to evaluate in that hypothesis space. In contrast, LLMs do not appear to include these steps. However, we believe that LLMs are implicitly doing a similar search, but their hypothesis space and specific hypotheses are buried in the weights, node biases (continuous parameters), and fixed architecture (e.g. transformer architecture [9]). Because LLMs represent a lot of their knowledge in the form of continuous parameters they can use gradient descent methods to improve their fit to the data. However, this form of knowledge representation means that it is extremely difficult for humans or machines to understand what has been learned since the language of weights and biases does not correspond to a language humans or machines typically use.

In our MEI approach, we expect that creative model space generation can represent and find any domain structure in the data that an LLM can, but in a form that is understandable. In the MEI method, when significant constraints are added to the constraint store they are not modified by subsequent learning. This means that the MEI method does not suffer from "catastrophic forgetting"–a well-known problem with DNNs. Also, learned knowledge in the MEI method can be transferred to a long-term knowledge store where it can be compared to what is already known to keep long-term knowledge consistent. Because the learned models are usually represented logically, it is then possible to use logical inference to check the consistency of new logical axioms against prior information also stored logically. In addition, the accumulated knowledge store resulting from extensive long-term ML can be used as a source of strong prior information for learning in new domains.

Another model representation language, used in Solomonoff induction [8], is to represent all models as an instruction set for a Turing machine. Data compression corresponds to finding the shortest instruction set that can reproduce the data. If the data is incompressible, then the shortest program

is the instruction "Print this data". The Turing machine model representation has the virtue that it is capable of representing any computable function, but finding the shortest instruction set is computationally infeasible. For this reason, Solomonoff induction is only used for theoretical analysis. Traditional ML, at the other extreme, typically only uses a single model space, and this may explain why they have performed poorly in comparison to LLMs.

5 Overfitting and Transfer Learning

A major issue in ML is the problem of avoiding *overfitting*. This is usually described as the problem that arises when models learned on one data set (the training data) are then used for prediction in another similar data set. If the models learned on the training set are overfitted, they fit the specifics of the training data that do not generalize, and thus their predictive accuracy on the new data declines. There are two different approaches commonly used in ML to avoid overfitting: cross-validation and regularization. Cross-validation trains on a subset of the data, but the accuracy of the model predictions is judged on the left-out data. Regularization instead penalizes the model complexity to prevent it from overfitting on the training data. The overfitting problem is closely related to the "transfer learning" problem: How can models learned on one data set be transferred to new data sets?

We take a different approach to ML that avoids the overfitting problem. In MEI, as described above, only constraints (models) that are significant relative to the training data are learned. The significance testing can be thought of as a form of regularization. These data-specific models can include components that readily generalize across related data sets, but can also include constraints that are specific to the training data. We avoid overfitting by applying MEI to each data set separately to get maximum predictability within each data set; we then take the set of models learned from these many data sets and apply ML to them by treating the learned models as data. In performing this type of meta-learning, only structures (models) that are predictive *across* data sets will be judged as significant in the meta-model. These meta-models can then be used as data for meta-meta-learning, and so on (hierarchical learning). These meta-models can be thought of as providing a bridge between more specific domains where only commonalities between the domains are represented in the meta-models.

For example, for multiple English texts, basic English concepts such as words, grammar, spelling, etc., would be learned as part of the meta-model across English texts, but author-specific structure, such as style, word choice, sentence length, etc., would not generalize to the meta-model because they vary with each text. Generalizations between languages would be much weaker, with concepts such as using words and grammar being a common link, but with a high degree of commonality at the semantic level.

6 Models and Model Space Generation

The least understood part of the hypothesize and test cycle (Fig. 1) or its MEI equivalent is how new hypotheses are generated. Hypothesis generation in humans is a creative act. Also, humans have a large, general understanding of the world which guides their choices about which hypotheses are likely to be more promising. These abilities are difficult to replicate in AI systems. In our approach, all model spaces begin populated by several well-known, highly generalizable, statistical models such as N-grams, regression, curve fitting, supervised/unsupervised classification, etc. Also, a library of well-known probability distribution functions such as the Normal, Poisson, Multinomial, etc., can be used for representing likelihoods. While the applicability of these "general purpose" models is broad, there are some assumptions about the format of the data which determine their applicability. Using this "toolkit" the agent may form some initial hypotheses about the structure in the data. In our text example, using a first-order N-gram model, an agent could test the hypothesis that the characters are not equiprobable by finding the frequencies of each character in the text.

Entirely new models could be generated automatically by applying generalization rules to known models in the set of commonly used models. For example, in n-gram models, a 3-gram is represented by a sequence: $\forall_i C_i C_{i+1} C_{i+2}$ where i is the start index. New models could be generated by considering sequences defined by: $\forall_i C_i C_{i+2} C_{i+4}$–i.e., characters taken 2 at a time. Generalization rules typically replace constants with variables of the same type, or other constants as in this example. Whether

all possible models could be generated by applying generalization rules to the initial set of models is unknown.

7 Summary and Conclusions

In this paper, we assume that machine learning can be cast as a search through model space to find models that best describe the data. Here, "best" means maximally compresses the data. This compression is achieved by performing an incremental search through models in the current space, and testing whether proposed models are significant relative to their MaxEnt value (MEI). Any such significant models (constraints) reduce the entropy of the data. These constraints are accumulated and used in computing future MaxEnt values. We give a procedure for performing this open-ended incremental search based on an example given by Jaynes [5]. Data compression is formalized by a method known as Minimum Message Length, and this in turn is a discretized version of Bayesian inference using MaxEnt priors and likelihood functions.

The MEI method described here is designed to automate the hypothesize and test cycle, allowing machine learning to operate autonomously. When such an autonomous learning method is let loose on masses of data available on the internet, it should be able to achieve performance comparable to or better than that shown by current LLMs. The explicit/declarative representation used in our MEI method allows learned knowledge to accumulate in a form that makes it easy to reason about and check for consistency. It is also a useful source of prior knowledge to help guide the search for further knowledge. Also, this declarative representation allows new model hypotheses to be generated beyond those traditionally used in machine learning.

References

[1] Peter Cheeseman. "A method of computing generalized Bayesian probability values for expert systems". In: *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 1*. 1983, pp. 198–202.

[2] Danielle Denisko and Michael M Hoffman. "Classification and interaction in random forests". In: *Proceedings of the National Academy of Sciences* 115.8 (2018), pp. 1690–1692.

[3] Peter Grunwald. *The minimum description length principle and reasoning under uncertainty*. University of Amsterdam, 1998.

[4] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[5] Edwin T Jaynes. "Where do we stand on maximum entropy?" In: *The maximum entropy formalism* (1979).

[6] Michael Irwin Jordan. *Learning in graphical models*. MIT press, 1999.

[7] Claude E Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[8] Ray J Solomonoff. "A formal theory of inductive inference. Part I". In: *Information and control* 7.1 (1964), pp. 1–22.

[9] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).