ABSTRACT THEORIES

ZENNA TAVARES*

Abstract. Theories and programs are expressive objects. It expressiveness

Key words.

Theories impose constraints on possible worlds; they declare what is possible, or probable conditioned on evidence.

What;s the relationship between theories and programs?

Why are programs difficult to induce?

They must be represented, and our choice of representation defines what space of theories can be considered, and the tractability of inference. Humans possess, and consistently execute a remarkable capacity for constructing theories which are abstract, combinatorially complex and recursive. Physical theories such as the law of gravitation invole looping over an abstract quantity of objects, psychological theories of mind and game playing may involve recursively considering the intentions of agents, and generative theories of vision may depend on graphical simulators. The composition of the kind of theories developed both intuitively and delibaratively, and formally and informally. This analogue has led to the hypothesis that sufficiently rich representation for theories is something like, or explicitly is a program. The universality afforded to theories as programs does not come without cost, I intend to focus my research

The problem stems from the fact that theories are programs

Theories and Programs. What's the relationship between theories and programs? In the Josh sense a In a more explicit sense Generative models are one kind of theory, which seek to. Probabilstic programs are emerging as a particularly expressive representation for expressing generative models. A program is a theory, and a computation is a deduction from that theory. The syntactic form of a program is indicative of its semantics only to an intelligent reasoner; the mapping is complex. This, and the inherent discreteness of logical structures contributes to what is refered tp discontinuity. It should be made explicit though, that any such notion assumes a metric on the space of programs, presumably evaluated syntatically. Together these have prohibit the use of simple based which perform well in other domains The suggested alternatives have been to transform the search space, such that a natural metric without the adverse properties can be defined, to smooth the landscape.

Constructive Bias. Bias is necessary for inference, machine learning and search. All learning algorithms employ some mechanism to restrict the hypothesis space, or prefer one hypothesis over another. These mechanisms are collectively known as inductive bias. It is illuminating to dissect from a bias its declarative component, which describes the manner in which the hypothesis space is being weighted. In this declarative sense, bias is equivalent to a number of terms, in particular constraints. It is in the other properties of bias, which cause these terms to differ, and are presumably the reaso we have different words for them. This exemplifies one axes of variation of bias, that of constructiveness versus restrictiveness, which I belive lies at the heart of the.

*Massachusetts Institute of Technology

1. Declarative vs Imperative. Formulating prolbems in terms of cosntraints enables a natural, declarative formulation of what must be satisfied, without having to say how it should be satisfied. In a declarative sense, a constraint is synonym of bias it i a restriction of the hypothesis space In the constraint processing literature, the problem is defined more finely grained; a problem is deco,posed to a set of variables, each drawing a vlaues from some domain. A constrant is simply . uar

Inferring Equalities. Theories are often formally expressed as equalities and inequalities: logical assertions on the equivalence of two expressions. An expression is a finite combination of symbols belonging to some formal language; for this we following convention in using s-expressions, nested list datastructure popularised by Lisp. as its dual purpose as a programming language permits evaluation of an expression, and hence the equality in entireity.

Given a dataset of variables $\mathcal{D} = (x_1, x_2, ..., x_n)$, each composed of N corresponding real-valued datapoints $x_i \in \mathbb{R}^N$, we aim to discover a system of equalities and inequalities which govern the data. To govern in this sense is subtle. We must relax the hardness of logical constraints in order to deal with real world data corrupted with noise. We must also introduce some inductive bias to prefer some equalities over the infinite number of alternatives.

Generalised difference and abstract constraints are the two methods we wish to follow.

Difference reduction. A common technique employed in the sciences and regression analysis is to inspect the residuals, or fitting error.

Considering a variable x, we call the additive error with respect to model $m \in \mathcal{M}$, the dataset $R_+ = (r_i | r_i = y_i - m(x_i))$. Note that least squares regression seeks to minimise $S = \sum_{i=1}^{n} r_i^2$. We would like to generalise this in two ways, firstly to account for n-ary conditions, and secondly to generalise error from additive to a general function.

One central procedure is to establish a template form, and explore ways in which this form would be modified in order to account for its deficiencies. Consider a set of characteristic features C, each a predicate on the data, $f \in C : \mathbb{R}^n \rightarrow true, false$. Examples might be: periodic? smoothly varying? constant-period?. We then define a conditional probability table for each model, $P(model|C)$. The first objective then is to learn this probability table. We then ask hypotheticals, perturbations to our conditioned attributes to create modified C', and observe the affect on p(model—C). For perturbations which increase the $p(model|C)$, e.g. if the data were smoothly varying then a sinusoid becomes a more likely place to start, so then, what do we have to do to change out data to make it smoothly varying, or what do we have to do to our model to make it not smoothly varying. What change do we need to make to our model, such that this characteristic is no longer.

Formally the task is to

Define or learn a set of predicates C. For each model m, learn a conditional probability table $p(m|C)$ Considering C and perturbed function C', let $C \rightarrow C'$ as the set Learn the perturbation function

Constraint. dd

Algorithm. At thie highest level, the algorithm constructs equations by executing transformations on sets of equations; it can be viewed as traversing the powerset of equations. Transformations are data driven and are based on t

Recognition (or proxy thereof):, the idea that abstract. Extension: First, the idea that abstract.

In words, we attempt to recognise some general structure in the data. Currently this is implemented by iterating over the set of models. Then we attempt to extend a candidate model, by first proposing an extension