Chapter 1

Introduction

Learnibility and computability are rarely considered in the same class. Computation concerns algorithmic solutions to problems whose answers are mathematically determinate after the input is given and, hence, belongs to the realm of deductive reasoning, where the premises infallibly guarantee the truth of the ensuing conclusion.

Learnibility concerns something more risky and less definite. An intelligent system has to “get the gist of” or “catch on to” a rule or concept after seeing a few instances of it, without waiting to see all of the infinitely many possible instances. Hence, it must “leap” to possibly false conclusions. If computability idealizes step-by-step algorithmic reasoning to a mathematically determinate conclusion, learnibility idealizes what can be accomplished by guesswork based on a finite chunk of potentially infinite experience.

Traditionally, philosophy has distinguished rather sharply between *deductive reasoning*, in which the conclusion is in some sense already contained in the premises, from *inductive* reasoning, in which the conclusion extends the premises. Deductive reasoning is usually associated with mathematics, proofs and algorithms. Inductive reasoning is usually associated with empirical science, statistics, and probability. Deduction is infallible and final. Induction is fallible and revisable. In almost every respect the two types of reasoning seem to be quite different, suggesting that the logic of induction must be quite different from ordinary deductive logic.

Traditionally, philosophers and many others have viewed logic and computability as the appropriate tools for understanding the foundations of mathematics and have viewed statistics and probability as the appropriate approach to inductive reasoning. It seems very intuitive to say, after all, that algorithms are impossible unless the right answer is uniquely determined and that, if it is not, then the best one can do is to achieve a kind of probabilistic confidence that some answer is right.

In this class, we will buck the tide of popular opinion by studying learnibility and computability together, in a unified mathematical theory of inductive and computational unsolvability. The basis for our approach will be computational learning theory, an interdisciplinary approach to inductive reasoning that embodies ideas from the theory of computability itself, along with allied ideas from analysis, topology, descriptive set theory and mathematical logic. The ideas aren't hard to understand, but they are abstract and usually aren't thought about in terms of empirical or inductive reasoning, so you will be asked to stretch your powers of abstraction a bit. You will also be asked to re-examine some very natural-sounding philosophical dogmas.

The basic idea behind computational learning theory is that empirical problems are like formal problems except that the inputs never stop coming in. Empirical methods are like algorithms except that they continue to receive inputs and to produce successive outputs forever. In both cases, the aim is to eventually end up with the right answer. In both cases, it may be impossible to find a procedure guaranteed to halt with the right answer. In both cases, there may nonetheless be a procedure that eventually arrives at a correct response after some number of mistakes and retractions of earlier answers. And in both cases, one can ask for the best sense in which a given problem is solvable. Looking at it this way, formal and empirical reasoning start to appear more similar than different.

Suppose we have in mind a series of ever weaker senses of success. The best such sense in which a problem is solvable characterizes its intrinsic difficulty or complexity. A system of degrees of complexity is called a complexity hierarchy. We will be concerned with complexity hierarchies throughout the course, and with how complexity determines the best approach to a problem. While thinking in terms of complexity is now routine in the case of computability, it is equally apt in the domain of learning and inductive method, and the principal novelty of the class will consist in explaining how.

The aim of the course is to provide a solid introduction to the classical results in the theory of computability and to provide them with a deeper significance in terms of consequences for induction and learnibility. Mathematics and science are always most interesting when fruitful analogies carry deep results from one domain into a new and unexpected application. Our aim is to construct a systematic analogy between empirical and formal reasoning that illuminates at a deeper, mathematical level, the nature and prospects of both.