ID: 10673226

COURSE: INTRODUCTION TO AI

The first paragraph basically talks about the doubts and uncertainties people have about learning. The paragraph also elaborates more on how sure we think our learning algorithm can produce a hypothesis that will correctly predict the value for previously unseen inputs. A question was also asked in the form how do we know that the hypothesis h is close to the target function f if we don’t know what f is. Other questions like how many examples do we need to get a good h? What hypothesis space should we use? If the hypothesis space is very complex, can we even find the best h, or do we have to settle for a local maximum in the space of hypotheses? How complex should h be? How do we avoid overfitting? Emerged later and all these questions were examined in the rest of the paragraphs.

The question of how many examples are needed for learning was tackled first. This question was answered by referring us to the learning curve for decision tree learning on the restaurant problem (Figure 18.7 on page 703) that improves with more training data. Learning curves are useful, but they are specific to a particular learning algorithm on a particular problem. Are there some more general principles governing the number of examples needed in general? Questions like this are addressed by computational learning theory, which lies at the intersection of AI, statistics and theoretical computer science. We also found out that the underlying principle is that any hypothesis that is seriously wrong will almost certainly be “found out” with high probability after a small number of examples, because it will make an incorrect prediction. Thus, any hypothesis that is consistent with a sufficiently large set of training examples is unlikely to be seriously wrong: that is, it must be probably approximately correct. Any learning algorithm that returns hypotheses PROBABLY APPROXIMATELY CORRECT PAC LEARNING that are probably approximately correct is called a PAC learning algorithm; we can use this approach to provide bounds on the performance of various learning algorithms.

Finally, the last paragraph of the whole chapter made us aware that PAC-learning theorems, like all theorems, are logical consequences of axioms. When a theorem (as opposed to, say, a political pundit) states something about the future based on the past, the axioms have to provide the “juice” to make that connection. For PAC learning, the juice is provided by the stationarity assumption introduced on page 708, which says that future examples are going to be drawn from the same fixed distribution P(E) = P(X, Y ) as past examples. (Note that we do not have to know what distribution that is, just that it doesn’t change.) In addition, to keep things simple, we will assume that the true function f is deterministic and is a member of the hypothesis class H that is being considered. The simplest PAC theorems deal with Boolean functions, for which the 0/1 loss is appropriate. The error rate of a hypothesis h, defined informally earlier, is defined formally here as the expected generalization error for examples drawn from the stationary distribution: error(h) = GenLossL0/1 (h) = x ,y L0/1(y, h(x)) P(x, y) . The error(h) was further explained in the rest of the chapter and the Sample Complexity was also explained. Lastly the hypothesis space H needed to be restricted in a way in order to obtain real generalization to unseen examples.