Improved PAC bounds

John Langford

Problem: Traditional PAC model bounds provide an elegant theory which bounds the number of examples required in order to guarantee that the emperical error is near to the expected error of new examples drawn from some (unknown) distribution over examples. Unfortunately, the PAC model bound is so loose that it is useless in practice.

Impact: Deriving tighter bounds for learning algorithms can have a significant impact on the way in which machine learning is done. Ideally, we'd like to calculate a PAC model type bound which was tight enough to use inside the program so as to causing halting before overfitting.

State of the Art: The state of the art in PAC model bounds is relatively weak. The PAC bounds are too weak to be effective. One current approaches involve using a hold out set to estimate the amount of overfit. This approach is unsatisfactory because data which could be used for training is not. Another common approach is "cross validation" which is essentially the holdout process run many times with a different set held out each time. From these many runs, an estimate of when overfitting occurs is made, and then the learning algorithm is applied once more to the full dataset stopping at the point where overfitting would occur. Cross validation is unsatisfactory because it is compute intensive and there is no strong theoretical justification for the process.

Approach: My approach has started with the observation that many learning algorithms proceed with a sequence of small choices to arrive at a final choice. A chernoff bound on the probability that the emprical loss of a hypothesis differs from the expected loss under the distribution giving examples applies for all $\epsilon > 0$. By tuning (a. priori but not explicitly) these ϵ for each hypothesis, we can arrive at a bound which is calculatable a. posteriori by looking at the trace of this running program.

Future Work: I'm interested in extending this work with both experimental results and more theoretical extensions. I've worked with Chuck on the experimental side and Avrim on the theoretical side to flesh out this idea.

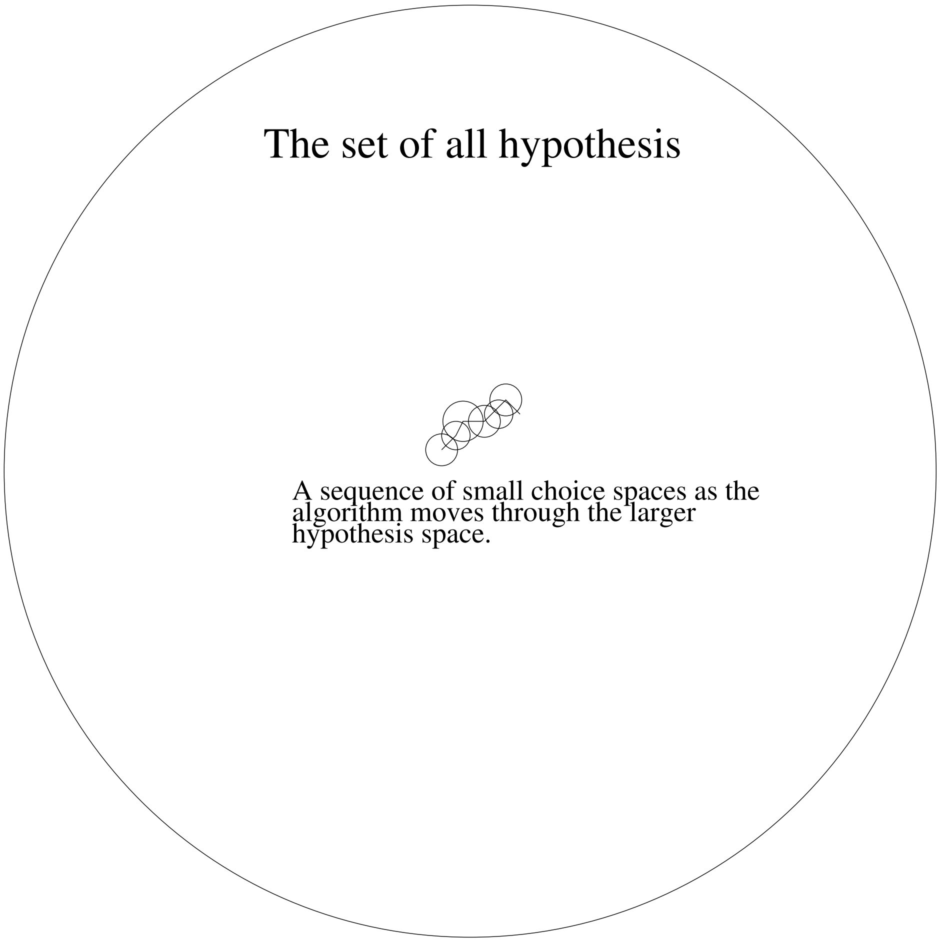

Figure 1: An algorithm which proceeds by making a series of small choices.