

# A Too-Clever Ranking Method

*Tim Oates*

I was a graduate student at a time when C4.5 ruled the machine-learning roost. I developed what I thought was an extremely clever method for detecting “bad” training instances. Each instance was scored, and those with the lowest scores could be removed before running C4.5 to build a decision tree with the remainder. I ran an experiment in which I removed the bottom 10 percent of the instances in a University of California, Irvine (UCI) data set. The resulting tree was smaller and more accurate (as measured by 10-fold CV) than the tree built on the full data set. Great! Then I removed the bottom 20 percent of the instances and got a tree that was smaller than the last one and just as accurate. At that point I had the feeling that this was going to make a great paper for the International Conference on Machine Learning (ICML).

So I kept going, removing an additional 10 percent of the instances at each step, getting smaller trees with no loss in accuracy. However, when I removed 80 percent of the instances, and was still getting the same result, I realized I had a problem. There was no way that 80 percent of the instances in any of the reversed UCI data sets were “bad.” After some time I realized I should run a control condition. What happens if I remove randomly selected training instances? Shockingly, I got the same results. The more randomly selected training instances I removed, the smaller was the resulting tree, with no loss in accuracy. My extremely clever ranking method was no better than a random number generator! After getting over the initial shock, I decided, with David Jensen, to pursue this more carefully with a larger sample of data sets. We found that this phenomenon was pervasive, both with respect to data sets and decision tree pruning mechanisms. We wound up writing papers on this topic that were published at the ICML, AAAI, and Knowledge Discovery in Datamining conferences, all because a surprising negative result made us look hard at what was going on.

**Tim Oates** is an associate professor in the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County, and he is director of the university’s Cognition, Robotics, and Learning Laboratory. He received his Ph.D. from the University of Massachusetts Amherst in 2001 and spent a year as a postdoc in the MIT AI lab. His research interests include the sensorimotor origins of knowledge, language learning, grammar induction, automated development of representation.

2. For a more in-depth look at MCL and the motivation behind it, see Anderson and Perlis (2005).
3. In the case of logic-based domains, an anomaly often takes the form of a direct contradiction,  $E$  and  $\neg E$ . This is the case, for instance, not only in the nonmonotonic reasoning domain, but also in the natural language domain discussed in this article. For these, we employ active logic (Elgot-Drapkin and Perlis 2006), a time-sensitive infer-

ence engine specifically designed to allow an automated agent to reason in real time about its own ongoing reasoning, noting direct contradictions rather than inadvertently using them to derive all sentences.

4. Active logic for reason enhanced dialogue.
5. When the tank is destroyed, it reappears at a random location on the map.