

1. Title: Tic-Tac-Toe Endgame database

2. Source Information

- Creator: David W. Aha (aha@cs.jhu.edu)
- Donor: David W. Aha (aha@cs.jhu.edu)
- Date: 19 August 1991

3. Known Past Usage:

1. Matheus,~C.~J., \& Rendell,~L.~A. (1989). Constructive induction on decision trees. In {\it Proceedings of the Eleventh International Joint Conference on Artificial Intelligence} (pp. 645--650). Detroit, MI: Morgan Kaufmann.
 - CITRE was applied to 100-instance training and 200-instance test sets. In a study using various amounts of domain-specific knowledge, its highest average accuracy was 76.7% (using the final decision tree created for testing).
2. Matheus,~C.~J. (1990). Adding domain knowledge to SBL through feature construction. In {\it Proceedings of the Eighth National Conference on Artificial Intelligence} (pp. 803--808). Boston, MA: AAAI Press.
 - Similar experiments with CITRE, includes learning curves up to 500-instance training sets but used _all_ instances in the database for testing. Accuracies reached above 90%, but specific values are not given (see Chris's dissertation for more details).
3. Aha,~D.~W. (1991). Incremental constructive induction: An instance-based approach. In {\it Proceedings of the Eighth International Workshop on Machine Learning} (pp. 117--121). Evanston, ILL: Morgan Kaufmann.
 - Used 70% for training, 30% of the instances for testing, evaluated over 10 trials. Results reported for six algorithms:
 - NewID: 84.0%
 - CN2: 98.1%
 - MBRtalk: 88.4%
 - IB1: 98.1%
 - IB3: 82.0%
 - IB3-CI: 99.1%
 - Results also reported when adding an additional 10 irrelevant ternary-valued attributes; similar _relative_ results except that IB1's performance degraded more quickly than the others.

4. Relevant Information:

This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row").

Interestingly, this raw database gives a stripped-down decision tree algorithm (e.g., ID3) fits. However, the rule-based CN2 algorithm, the simple IB1 instance-based learning algorithm, and the CITRE feature-constructing decision tree algorithm perform well on it.

5. Number of Instances: 958 (legal tic-tac-toe endgame boards)

6. Number of Attributes: 9, each corresponding to one tic-tac-toe square

7. Attribute Information: (x=player x has taken, o=player o has taken, b=blank)

1. top-left-square: {x,o,b}
2. top-middle-square: {x,o,b}
3. top-right-square: {x,o,b}
4. middle-left-square: {x,o,b}
5. middle-middle-square: {x,o,b}
6. middle-right-square: {x,o,b}
7. bottom-left-square: {x,o,b}
8. bottom-middle-square: {x,o,b}
9. bottom-right-square: {x,o,b}
10. Class: {positive,negative}

8. Missing Attribute Values: None

9. Class Distribution: About 65.3% are positive (i.e., wins for "x")