SE-trees Outperform Decision Trees in Noisy Domains

Ron Rymon*
Intelligent Systems Program, 901 CL
University of Pittsburgh
Pittsburgh, PA 15260
Rymon@ISP.Pitt.edu

Abstract

As a classifier, a Set Enumeration (SE) tree can be viewed as a generalization of decision trees. At the cost of a higher complexity, a single SE-tree encapsulates many alternative decision tree structures. An SE-tree enjoys several advantages over decision trees: it allows for domain-based user-specified bias; it supports a flexible tradeoff between the resources allocated to learning and the resulting accuracy; and it can combine knowledge induced from examples with other knowledge sources. We show that SE-trees enjoy a particular advantage over simple decision trees in noisy domains. This advantage manifests itself both in terms of accuracy, and in terms of consistency.

SE-tree-based Induction

- SE-trees first proposed as a systematic way to search a space of sets (Rymon, 1992). One way to view learning is as search for kernel (minimal) rules.
- An SE-tree can also be viewed as a generalizing decision tree induction a la Quinlan (1986) and Breiman *et al.* (1984), where nodes can possibly be expanded with *multiple* attributes (Rymon, 1993).
- The Algorithm (simplified version):
  - As in decision trees, recursively partition the training set until a rule qualification condition is met. Except in each node, this is done for all allowed attributes (termed that node's "View").
  - In each node, attributes in the View are scored by the attribute selection measure of choice, e.g., Information Gain, GINI Index, Chi-Square, and then expanded in that order.
  - Systematicity – never go back to higher-scoring attributes – ensures uniqueness of exploration.
  - Only most general rules are retained; if an old rule is subsumed by a new one, it is removed.
- Example (Figure 1):
  - W.l.o.g, suppose attributes are first scored in a lexicographic order, as marked next to the root. The root was thus expanded with all attribute-value pairs, in that order.

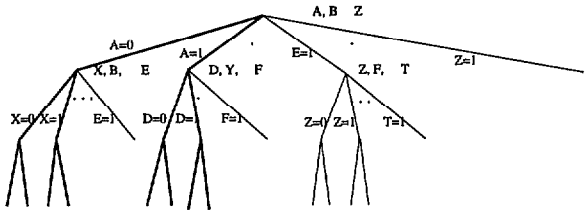

Figure 1: Improving Upon a Given Decision Tree

  - In every node, attributes in the View (i.e., those scoring worse at its parent) are re-scored according to same heuristic. For example, in the node {E=1}, attributes F-Z are re-ranked: Z, F, ... T; in the node {Z=1}, the *View* is empty.
- Main features of SE-tree framework:
  - Primary Decision Tree. In Figure 1, note the bold-faced arcs at the left side of the SE-tree. These are exactly same nodes that would have appeared in a decision tree utilizing same heuristic.
  - Union of Many Alternative Decision Trees. Due to its multi-branching, as SE-tree can be viewed as an economical representation of many alternative decision trees: all overlaps are uniquely represented, and only most general rules are retained.
  - User-Specified Exploration Policy. Complete SE-trees are often too large to be entirely searched. They are so searched by a user-specified *exploration policy*. One family of exploration policies begins with the primary decision tree, and then continues to other parts of SE-tree. This is a hill-climbing procedure, where the extent of exploration may depend on resources available. The exploration policy represents user-specified bias.
  - User-Specified Resolution Criterion. As in decision trees, new instances are classified along matching paths, except here there may be *multiple* such paths. User-specified criteria are used to resolve conflicts (note that same conflicts exist

*Parts of this work were supported by NASA grant MTPE-94-02; and funding from Modeling Labs.

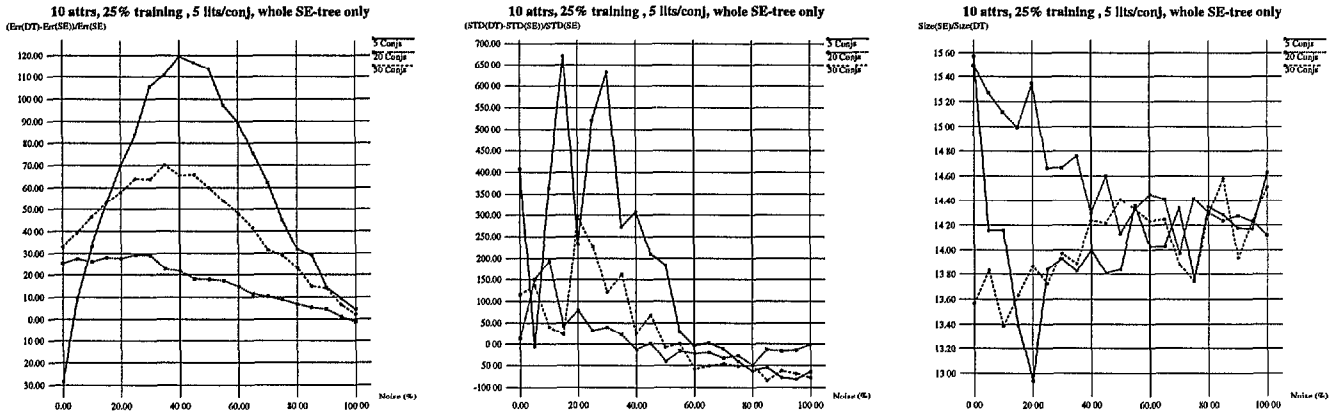

Figure 2: Varying #conj, #lits=5

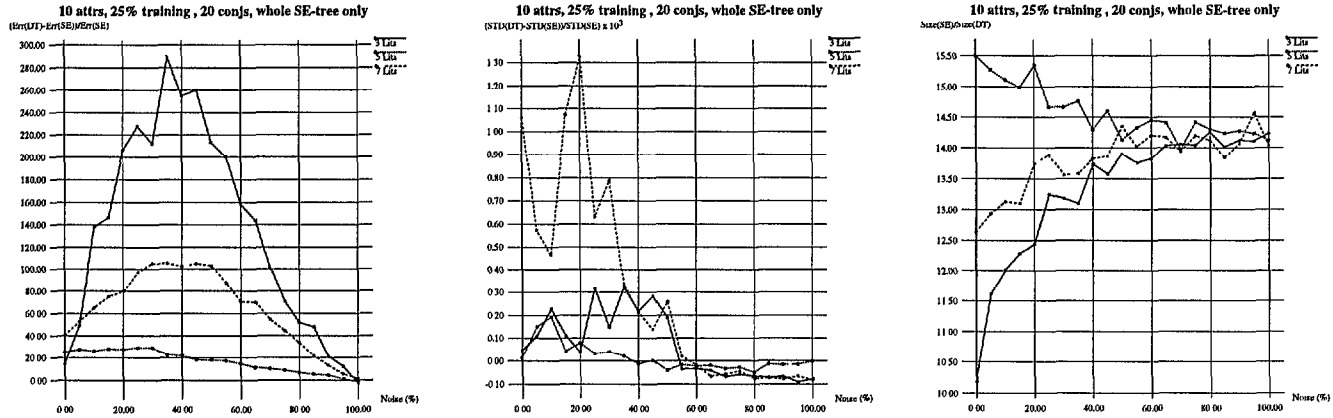

Figure 3: Varying #lits, #conj=20

between alternative decision trees). A resolution criterion can be general (e.g., simple voting), but may also represent domain-based bias.

- Implementation. This family of algorithms for SE-tree-based induction, including a variety of attribute-selection measures, as well as exploration policies and resolution criteria, is implemented in a program called SE-Learn (SE-Learn Home Page).

Hypothesis and Methods

- Hypothesis: SE-trees outperform decision trees in noisy domains.
- Intuitive Explanation: incorporating many alternative decision trees acts to reduce variance error.
- Method of Investigation: empirical, by testing on randomly generated artificial problem sets, with varying noise levels. We compare generalization accuracy of SE-tree and its primary decision tree.
- Using artificial problems, we can control generation and testing parameters. For the main experiment we use the following parameters:
  - Problem size: 10 binary attributes + binary class.
  - Training set size: 25% of the domain.
  - Complexity of target function: using DNF representations, diversity is achieved by varying the number of conjunctions (#conj), and the number of literals per conjunction (#lits).
  - Distribution of problems: Uniform over a given choice of #conj and #lits.
  - Choice of decision tree program: Information Gain (ID3) is used to select splitting attributes.
  - Choice of parametrization for SE-tree framework:
    - Exploration policy and extent: Complete SE-trees only (conservative assumption).
    - Resolution criterion: Simple voting (again a conservative assumption).
  - Pruning: Unpruned trees.
- In subsequent experiments, we test sensitivity to important assumptions.

Main Experiment

- Two experiments: varying #conj, and varying #lits.

As aside, it appears as if 20-conj 5-lits functions are more complex than other variations.

- Noise modeled as randomly assigned classes, 0-100%.
- Test complete SE-tree vs. primary decision tree.
- 30 different random problems tested per data point.
- Reporting:
  - Normalized reduction in average error rates: 100*(Err(DT)-Err(SE))/Err(SE);
  - Normalized reduction in variance: 100*(STD(DT)-STD(SE))/STD(SE);
  - Cost as complexity ratio: Size(SE)/Size(DT).
- Results (Figures 2, 3):
  - Generally SE-trees have lower error rates (as indicated by positive values).
  - At moderate noise levels, error reduction increases with introduction of additional noise. By definition, it must then drop back; at 100% noise, there is no information in training set.
  - SE-trees are more consistent at moderate noise levels, but less consistent at very high noise levels.
  - Complete SE-trees are more complex. However, they are typically explored partially. Generally, complexity ratios are higher for more complex functions. It appears that noise reduces the ratio for complex functions, and increases it for simpler ones.

Experimental Setting Variations

- Conservative assumption: performed with 20-conjs 5-lits functions, where the relative performance of the SE-tree was poorest: #conj=20 and #lits=5.

1. Alternative attribute selection heuristics (Figure 4). No apparent differences.
2. Training set sizes (Figure 5). Hypothesis still confirmed. However, lower absolute error reduction rates and higher complexity ratios are apparent with fewer training instances.
3. Statistically pruned vs. Unpruned trees (Figure 6). SE-tree still outperforms decision tree. However, in pruned trees, we find smaller error reductions, and larger variations. In pruned trees, complexity ratios are smaller, and drop quickly with noise.

Related Work and Discussion

- Tree averaging. As a joint representation for many alternative decision trees, the SE-tree-based framework bears resemblance to the tree averaging approach taken by Kwok and Carter (1990) and Buntine (1994), as well as Breiman's (1994) bagging.
- Reduction of Variance Error. Dietterich and Kong (1995) distinguish statistical bias from variance error. They indicate that "one important source of variance in C4.5 is the fact that the algorithm must choose a single split at each node". Noisy domains are characterized by added variance, and SE-trees may relieve some of it.

  Dietterich and Kong conclude that "some method is needed for converting a combination of trees (or other complex hypotheses) into a smaller, equivalent hypothesis. These trees are very redundant; how can we remove this redundancy, while still reducing bias and variance?". SE-trees may represent one step in that direction.
- Bootstrap Aggregation. In analyzing where bagging works, Breiman too points to domains exhibiting great variation between alternative classifiers. SE-trees provide an effective way to consider alternative decision trees without the loss of information incurred when subseting the training data.

Conclusion

1. In the presence of noise, a complete SE-tree typically outperforms its own primary decision tree. Results were replicated with different heuristics.
2. At moderate noise levels, the SE-tree advantage generally *increases* with additional noise.
3. At moderate noise levels, the SE-tree is also more consistent.

References

Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees.* Wadsworth, Belmont.

Breiman, L., Bagging Predictors. *Technical Report 421*, Department of Statistics, UC Berkeley.

Buntine, W., Learning Classification Trees. *Artificial Intelligence Frontiers in Statistics*, D. Hand (Ed), Chapman and Hall.

Dietterich, T. G., and Kong, E. B., Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms. Technical Report, Department of Computer Science, Oregon State University.

Kwok, S., and Carter, C., Multiple Decision Trees. *Uncertainty in Artificial Intelligence*, 4, pp. 327-335.

Quinlan, J. R., Induction of Decision Trees. *Machine Learning*, 1(1):81-106.

Quinlan, J. R., *C4.5: Programs for Empirical Learning.* Morgan Kaufmann, San Francisco, CA.

Rymon, R., Search through Systematic Set Enumeration. *Third International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge MA, pp. 539-550.

Rymon, R., An SE-tree-based Characterization of the Induction Problem. *Tenth International Conference on Machine Learning*, pp. 268-275, Amherst MA.

Rymon, R., Home page for the SE-Learn software. http://www.isp.pitt.edu/~rymon/SE-Learn.html.

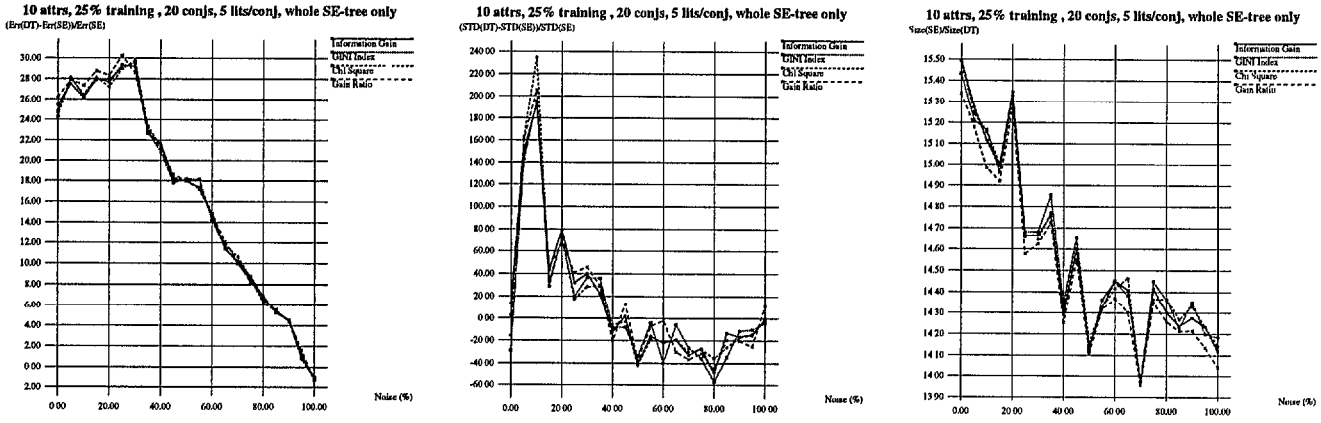

Figure 4: Various Attribute-Selection Heuristics

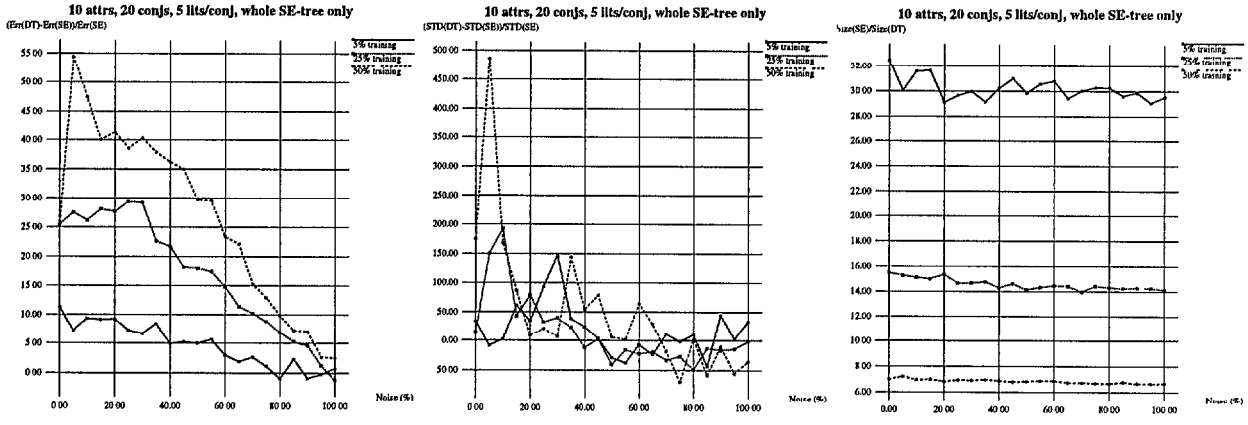

Figure 5: Various Training Set Sizes

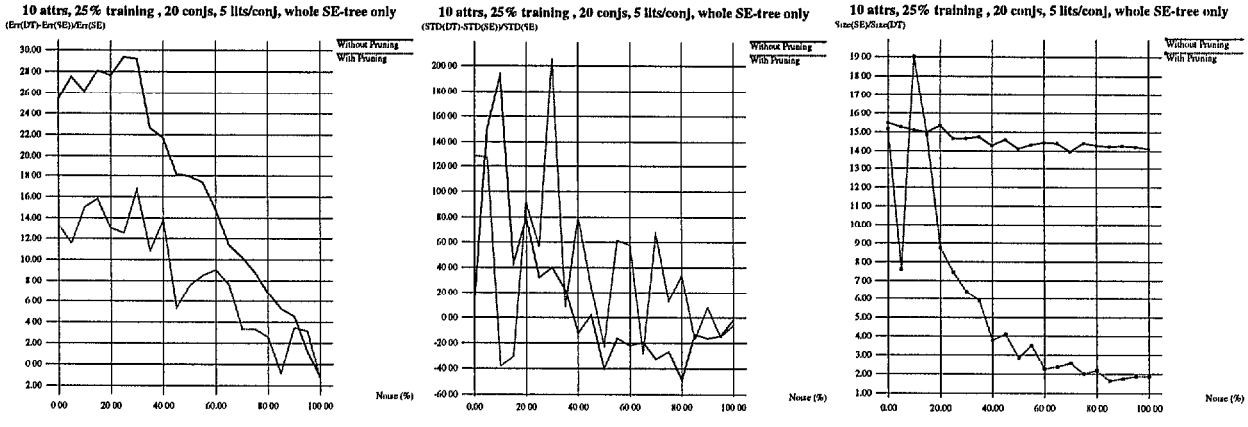

Figure 6: With and Without Pruning