Motivation

Ensemble methods for supervised machine learning have become popular due to their ability to accurately predict class labels with groups of simple "base learners." Many ensemble methods are computationally efficient, but offer little insight into the structure of a dataset. We consider an ensemble technique that returns a model of ranked rules. The model accurately predicts class labels and has the advantage of indicating which parameter constraints are most useful for predicting those labels.

Problem Formulation

Suppose we are given a dataset \( S = \{y_i, x_i\}_{i=1}^N \) where the label \( y_i \) is either +1 or -1 and \( x_i \) is a vector \((x_{i1}, \ldots, x_{ik})\) of \( k \) features. We want to predict which class and unlabeled observation \( x \) came from.

We use the rule ensemble method developed by Friedman and Popescu (2003, 2004, 2005) to construct a function \( F(x) \) such that \( \text{sign}(F(x)) \) predicts the true label \( y \).

As we assume \( F(x) \) is a linear combination of base learners \( f_i \)

\[
F(x; a) = a_0 + \sum_{k=1}^K a_k f_k(x)
\]

and approximate the coefficients \( a \) by minimizing the risk of using \( F(x) \) on the sample of training data \( S \).

The \( l^1 \) / lasso penalty is added to the minimization problem to make a sparse and prevent overfitting the training sample. A sparse solution returns a model with fewer terms that is easier to interpret. As \( a \) is a vector of weights for each rule, we can also interpret which rules and coefficients are most influential.

Rules as Base Learners

- The Rule Ensemble Method uses rules \( r_k(x) \) as base learners \( f_k(x) \).
- Each rule is a node in a decision tree \( r_k(x) = \prod f(x_{j} \leq p_j) \).
- Rules define hypercubes in parameter space and evaluate to 1 if an observation is in that cube, i.e. an observation's \( j \)th attribute abides by the rule's constraint \( p_j \).
- Rules are diverse, quick to build, and allow interactions.

Calculation of Coefficients

**PATHBUILD:**
Friedman & Popescu 2004
Uses a constrained gradient descent method to approximate \( a \) by solving

\[
\min_{a} \frac{1}{N} \sum_{i=1}^N L(a_0 + \sum_{k=1}^K a_k f_k(x_i), y_i) + \lambda \sum_k |a_k| \]

where \( L() \) is ramp loss error. Initializes all coefficients to zero and at each iteration updates only coefficients \( k \) that have large enough gradient \( \{ k : |\langle a_k(X, a^*) \rangle \rangle \geq r = \|\langle a_k(X, a^*) \rangle \|_2 \} \).

Constraint parameter \( \tau \) is in \([0, 1]\).

**Fixed Point Continuation:**
(Hale, Yin & Zhang 2007)
Uses a shrinkage operator to approximate \( a \) by solving

\[
\min_{a} \frac{1}{N} \sum_{i=1}^N \left( a_0 + \sum_{k=1}^K a_k f_k(x_i) - y_i \right)^2 + \lambda \sum_k |a_k|.
\]

This problem is a reformulation of the \( l^1 \)-penalized regression problem. It uses the \( l^2 \) norm for a loss function and the sparsity is controlled by \( \mu > 0 \).

Reduce 39 to 21 features

Using the magnitude of a coefficient to indicate the importance of a rule, the rule ensemble method indicated that 21 out of the original 39 attributes were more influential than the rest. Retraining and testing with the restricted set of attributes gave a lower overall error rate which indicates that the rule ensemble method successfully identified important attributes in the dataset.

We would like to thank Sean Peisert and Peter Nugent for their extremely valuable comments and suggestions. This research was supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231