ANTERUS

Genetic Algorithms and Evolutionary Programming

The Intelligence Factory supports both genetic algorithms and evolutionary programming. These capabilities are tightly integrated with the knowledge management and intelligence discovery capabilities are used to generate initial model structures, to tune models, and to generate candidate rules for a model. But genetic and evolutionary algorithms can also be used by model designers to solve complex, multi-objective, inter-dependent models. A Genetic Tuner can be used to used to optimize the configuration of the discovered knowledge base (1) thus producing a new, highly effective mode. You can also inject the discovered rule into an existing knowledge base (2). The Genetic tuner can then be run in tournament mode (3) to optimize the combine client and discovered rules. Along with this module is the Anterus Data Explorer GUI, a comprehensive data mining, knowledge discovery, and KB management environment.

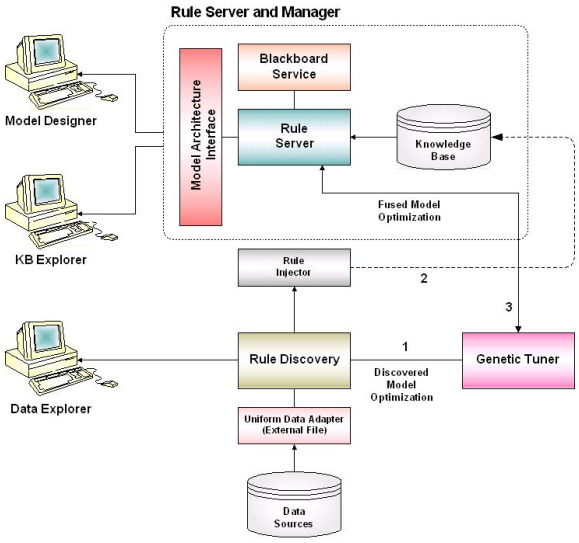

Figure 1. Knowledge Discovery and Optimization

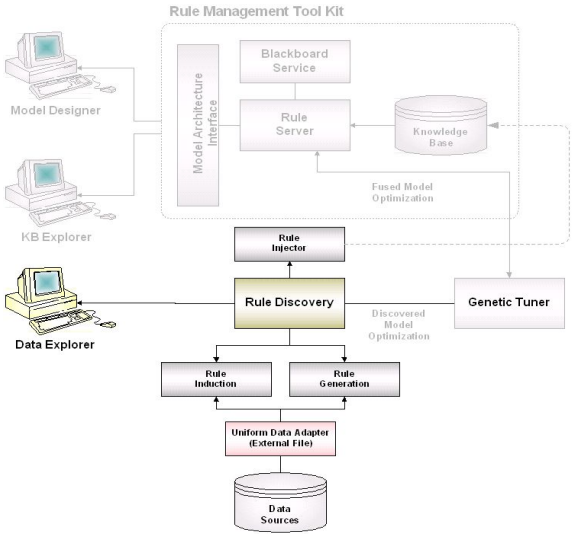

Figure 2. Rule Generation with Evolutionary Programming

The *Rule Generation* engine (Figure 2) uses Evolutionary Programming (EP) techniques to generate and test candidate rules against an objective function. The evolutionary programming approach employs a sophisticated genetic algorithm to "breed" and test new rules, combining rules that have higher and higher predictive power. Unlike the rule induction approach, which derives rules from patterns in the data, the rule generation approach randomly assembles rules and tests them against the data.

ANTERUS

Features:

Intelligence Factory Applications

Design and Tuning of fuzzy models
Generation of rule models through Evolutionary Programming
Optimization of knowledge through genetic tournaments
Scheduling and planning of complex, multi-constraint problems
Evolution of parameters for real-time statistical learning
Deep data mining of high attribute data
Selection of model variables and expressions
Selection of regression coefficients
Sensitivity analysis
Goal seeking and categorization

Genetic Algorithm

Multi-Objective, Hierarchical genome structure
Bit, Real, Integer, and mixed chromosome
Packet DNA definitions
Multiple strategies for:
- cross-over and mating
- population retention and pruning
- mutation and mutation site selection
- objective function convergence

Fuzzy and crisp properties
Rule-based generation control and access
Solution checkpoints and restart

Evolutionary Programming

Generates nested *if-then-else* rules
Generates mathematical models
While and *for* looping
Calculated expressions and declaratives
Fuzzy and crisp expressions
Shared, re-usable code generation
Coupled to Genetic Algorithm for
- Fuzzy set generation and optimization
- model refinement
- coefficient discovery

Integrated statistical, mathematical, financial, time series functions
Recursive procedural structure