TSC

Christopher Mark Gore

`chris-gore@earthlink.net`

`http://www.cgore.com/research/tsc`

University of Missouri – Rolla
Department of Computer Science
Natural Computation Laboratory
Friday, November 2, AD 2007

— Thesis Committee —

- Dr. Daniel Tauritz (thesis advisor), Computer Science
- Dr. Ralph Wilkerson, Computer Science
- Dr. Ray Luechtefeld, Engineering Management

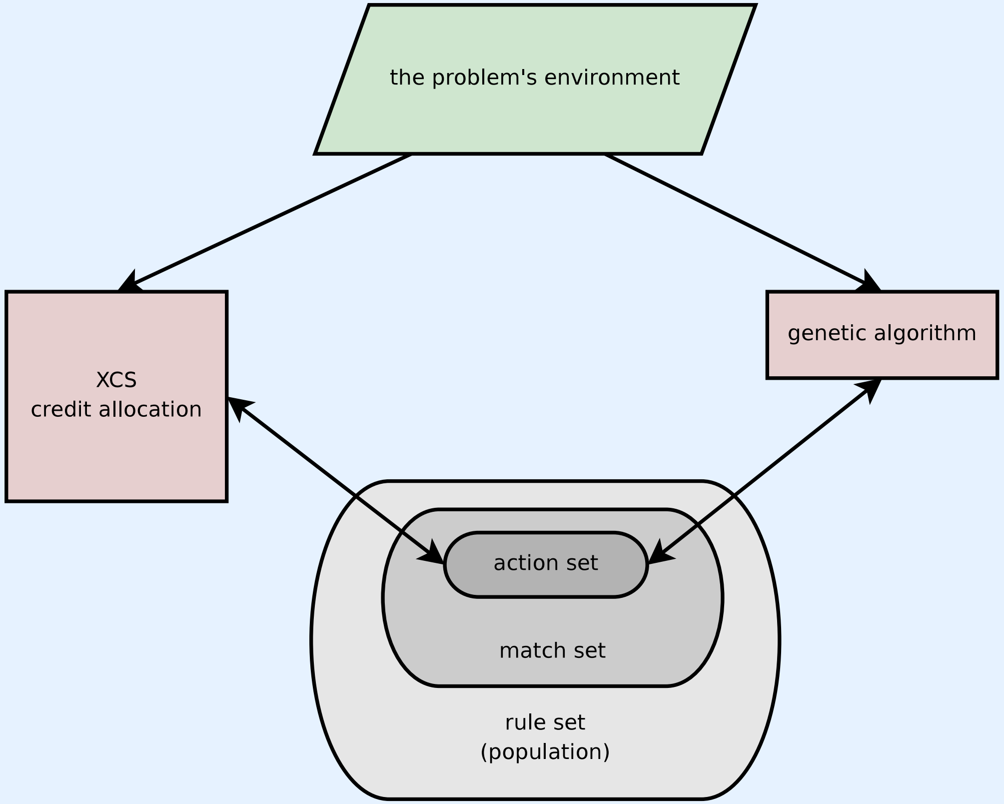
— The Evolutionary Process —

1. Initialize the population, either with randomly-generated or seeded candidate solutions.
2. Evaluate the fitness of each member of the population.
3. **repeat**
4. Select members of the population to act as parents. This is typically related to the relative fitness of the parents in some way.
5. Recombine the genetic material of the parents, producing offspring to be added to the population.
6. Mutate some or all of the newly-created offspring.
7. Evaluate the fitness of the offspring.
8. Select survivors from the current population or a subset thereof, often only the newly-created offspring, to remain in the next generation.
9. **until** some specified termination condition is satisfied

— Learning Classifier Systems —

- A learning classifier system (LCS) is a type of evolutionary algorithm (EA) in which a description of a current situation is used in an attempt to map that description to some classification or action.
- This is achieved through simulated evolutionary processes, where the population being evolved consists of various rules; our entire population forms a rule set, and we apply concepts from Darwinism to our individual rules.
- This is known in learning classifiers as the *Michigan approach*. TSC is a modification of XCSR, and both use the Michigan approach.
- The other primary method employed, where each individual is an entire solution and therefore a whole rule set is known as the *Pittsburgh approach*.

— XCS's Basic Structure —



— XCS's Rules —

Each rule r is now of the more complex form

$$r = (c, a, p, \epsilon, F, exp, ts, as, n), \tag{1}$$

where:

- c is the condition matched by the rule r, comprised of elements from some alphabet such as $\{0, 1, \#\}$, where $\#$ is the matching symbol, matching both 0 and 1.
- a is the action that the rule r recommends.
- p is the predicted payoff.
- ϵ is an estimate of the prediction error.
- F is the fitness used by the GA. It is vital that the fitness used by the GA is a measure of the *accuracy* of the rule, and not a measure of the *magnitude*.

- exp is the experience of the rule, a count of the number of times since this classifier's creation that it has belonged to the action set.

- ts is a time stamp of the last occurrence of a call to the GA in an action set that this classifier was a part of, as the generational number.

- as is an estimate of the average action set size this classifier has belonged to.

- n is the numerosity of this macro-classifier. This is how many traditional micro-classifiers this macro-classifier represents.

— XCSR —

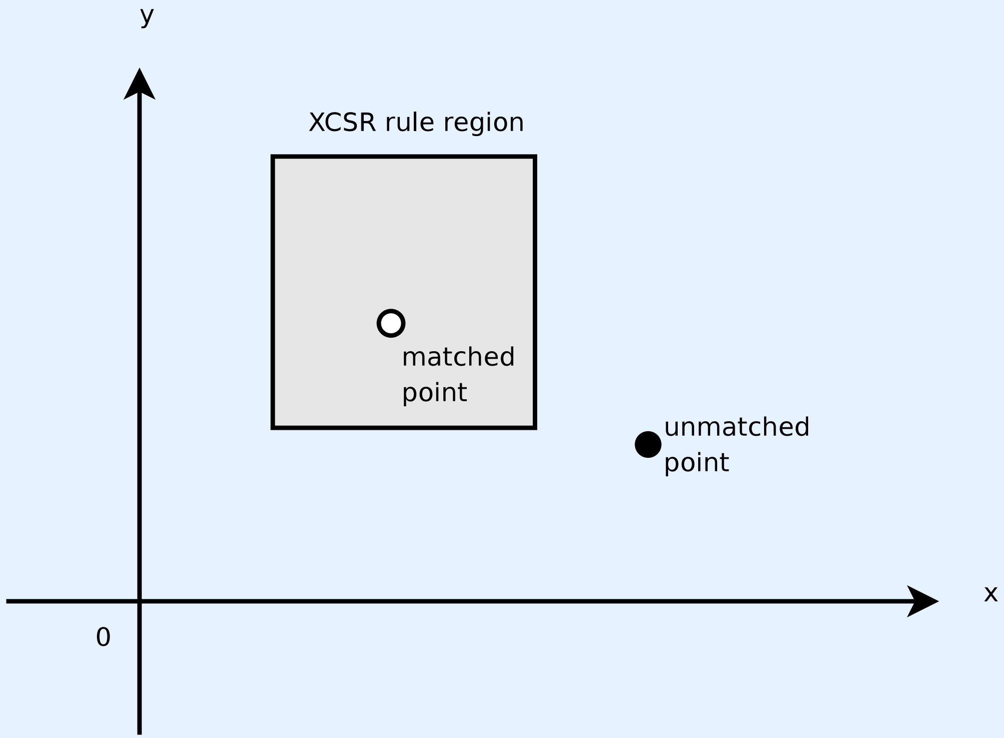
XCSR is XCS, but with real-valued predicates, which is preferable for most real-world problems. Intervals are represented as:

$$interval_i = \{lower_i, upper_i\}, \tag{2}$$

where x_i is matched by $interval_i$ iff

$$lower_i \leq x_i \leq upper_i. \tag{3}$$

— XCSR's Interval Rules —



— TSC —

- The primary focus here is TSC, is a learning classifier system derived from XCSR aimed at time-series analysis, a *time series classifier*.
- It attempts to learn useful information about the data presented to it in an online fashion and then make predictions about the next time step.
- With this information about the next time step, we can choose appropriate actions. For example, if TSC were to be used on weather data and forecast rain for tomorrow, we could bring umbrellas.

— Time Series Data —

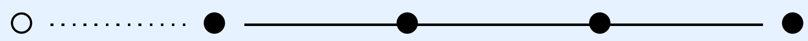
- We view the time series data as a time-ordered list of composite data structures. On this data we may apply entry-wise and path-wise transformations.
- For the stock data, the time series is trading information about a single stock.
- Each time step represents a trading day.
- Each day's data holds the opening, closing, high, low, and adjusted closing price, as well as the trading volume. These are the composite data elements for a single stock.

— TSC Rules —

- TSC rules are modified XCSR rules which are, in turn, modified XCS rules.
- The representation of a single rule is a collection of predicates; each predicate must match the current situation for the rule to match the situation. A single predicate consists of in initial and final position, a field selector ϕ, an operator ω, and a range pair consisting of a lower and upper bound $[l, u]$.
- The field selector ϕ is to be a lexical closure taking only one argument, which is the structure at the position $\{t, x_0, \ldots, x_{n-1}\}$.
- The operator ω is also a lexical closure, and is intended for classification purposes; all ω's must operate over a one-dimensional vector of data.

— Mutation —

The approach to mutation of the paths is to restrict the mutation of the line segment to the same line, only allowing the end points to move up or down along that line. In this method, the alteration of the line segment is minor, and therefore there is very little change in the actual information held by the path. This is exactly the sort of effect we wish in mutation: very small changes.



The lower and upper values of the range are altered, but limited by a maximum mutation parameter, and also limited to ensure that the current situation maintains its current classification under the classifier rule.

— Crossover —

We use a marginally-modified form of one-point crossover. Consider viewing the environment condition of rule as consisting of several predicates, each possessing an initial point A, a final point B, a lower bound l, an upper bound u, a field ϕ and an operation ω. We could choose to view this as a list of the form

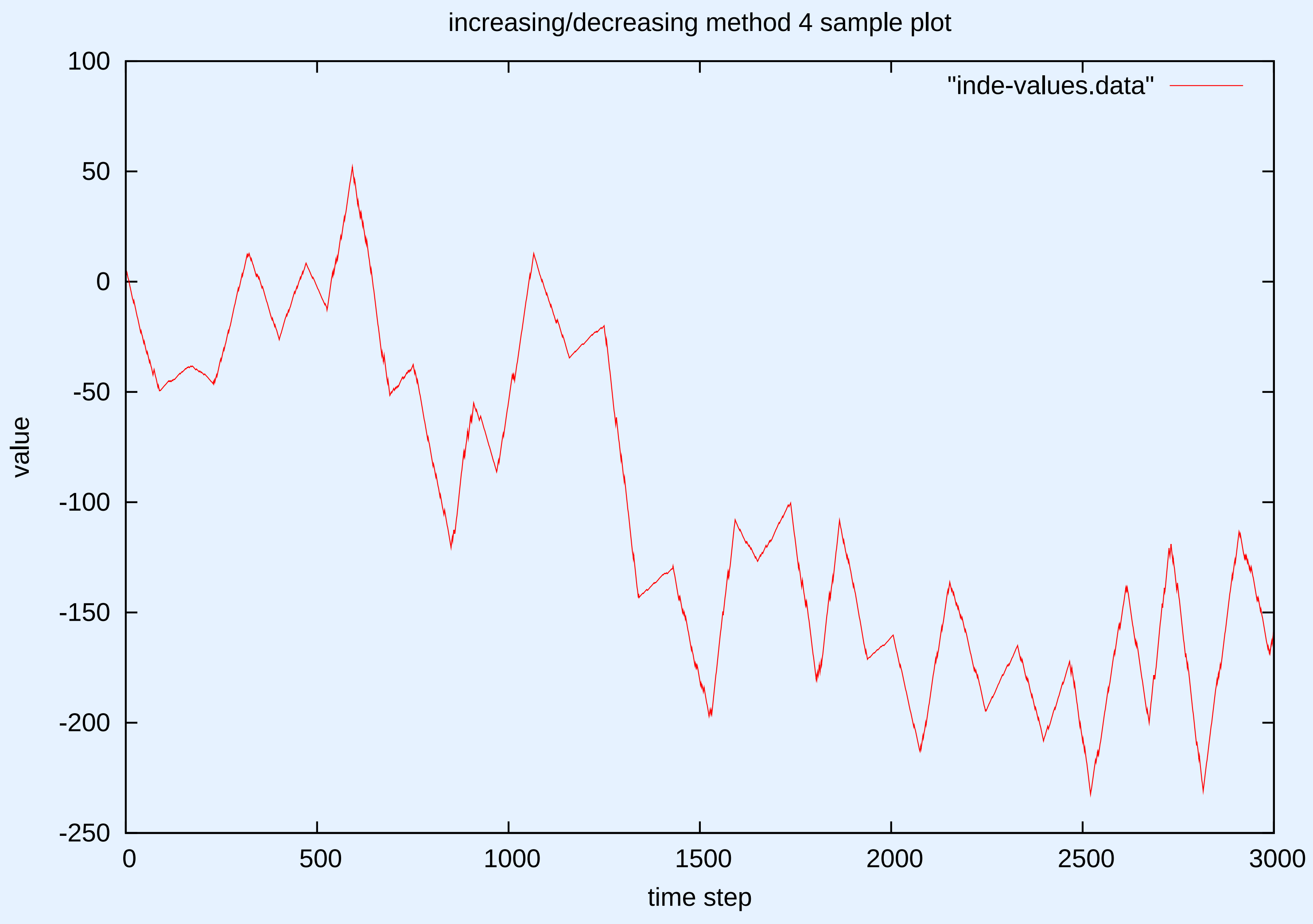
$$\{A_0, B_0, l_0, u_0, \phi_0, \omega_0, \ldots, A_{p-1}, B_{p-1}, l_{p-1}, u_{p-1}, \phi_{p-1}, \omega_{p-1}\} \quad (4)$$

where p is the number of predicates contained in the rule. Apply one-point crossover on two lists of this form, but insure that both lists break the predicates in the same way.

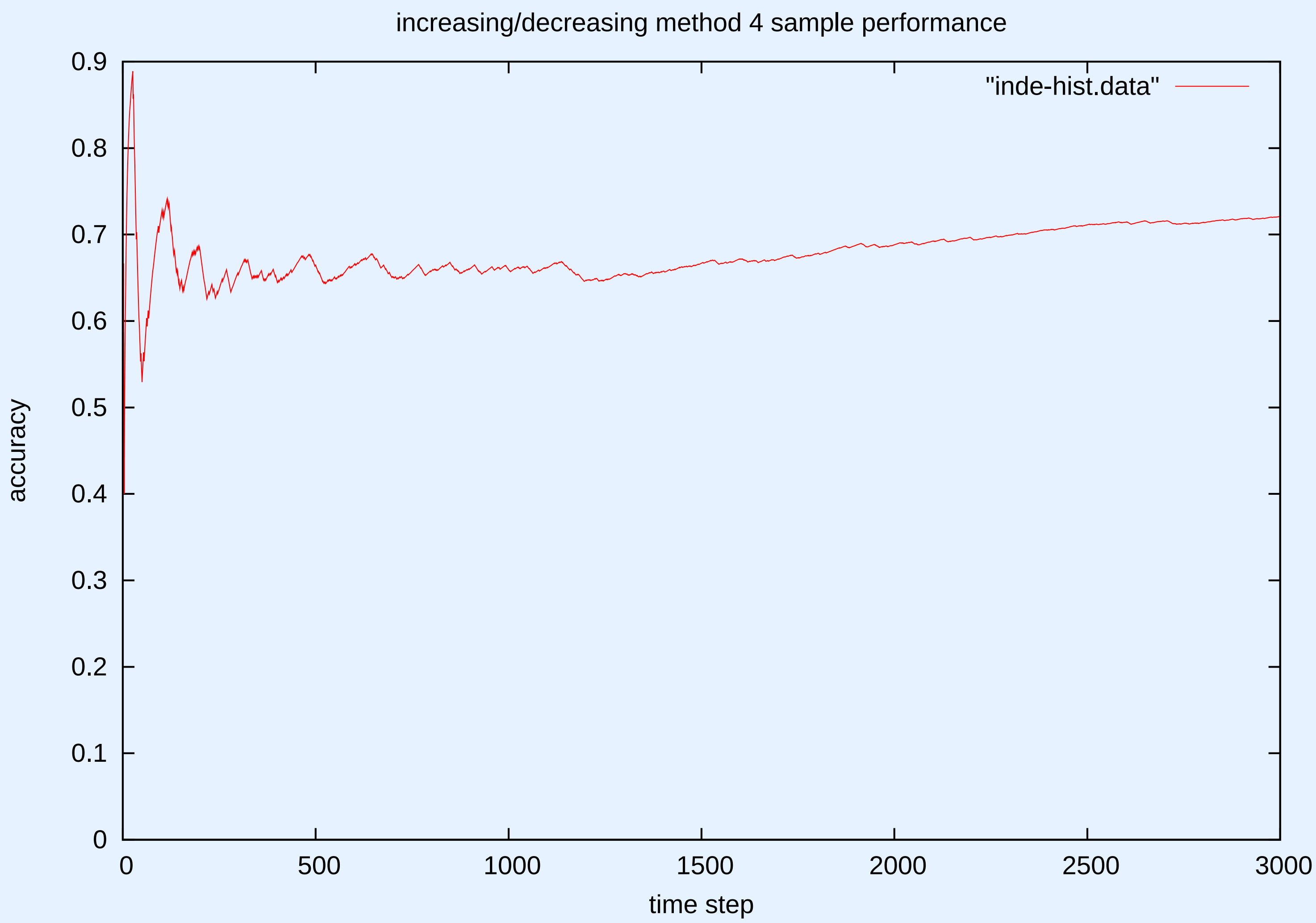
— A Simple Test —

Our simple test is designed to loosely resemble a traded entity.

- Randomly chosen slope,
- Random duration of the chosen slope,
- Random noise,
- Random direction (up/down).



increasing/decreasing method 4 sample plot
"inde-values.data"
100
50
0
-50
-100
-150
-200
-250
0
500
1000
1500
2000
2500
3000
time step
value



increasing/decreasing method 4 sample performance
"inde-hist.data"
accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
3000
time step

— The Stock Market —

The stock market is an excellent real-world test for any self-learning system. It is:

- very difficult,
- has a default score ($),
- has lots of associated data,
- and has a huge payoff for success.

— TSC Final Parameters —

parameter	value	return	B&H ratio
reward method	a_2	11.06%pa	0.67875
GA threshold, θ_{GA}	25	...	...
crossover probability, χ	0.9	11.85%pa	0.70797
mutation probability, μ	0.15	12.74%pa	0.74200
exploration probability, P_{explr}	0.3	13.23%pa	0.76145

— Results —

- The TSC returned an impressive 13.23%pa.
- TSC outperforms mutual funds (usually around 8%pa) and most especially a savings account (usually less than 1%pa).
- TSC never lost money in the long run for any trial.
- TSC underperformed buy-and-hold.
- A medeocre hedge fund manager can return at 20%pa or more and can be had for a salary of $500,000pa or less.

— Potential Future Work —

There are several opportunities for improvement on TSC. Some of these are obvious, and result from known simplifications and limitations of the current TSC system, and others are not yet known. The most obvious paths for future research with this TSC fall into four main tasks:

1. using more advanced ϕ's,
2. using more advanced ω's,
3. finishing the implementation of multidimensionality,
4. and applying the system to other real-world problems.

— Conclusions —

- TSC can learn complex time series data.
- TSC would be a useful system for real-world time series.
- There is room for improvement still with TSC.
- The stock market is too complex for TSC, and probably learning classifiers in general.

Questions?