

Incanto – Performance By Design

Incanto uses a number of methods to achieve the goals of high performance without losing the possibility of rich decisioning. The two major contributors to these goals are advanced rule functionality together with the integration of high speed fuzzy logic processing.

Incanto rule functionality

Incanto uses a modified decision tree in the form of a directed acyclic graph to perform rule processing, rather than a conventional inference based method. This allows an intuitive representation of the rules processing and easy amendment. It also has the advantage over inference based systems of both well-defined rule navigation and high performance. To mitigate the brittleness associated with conventional decision trees, **Incanto** modifies the structure to place the decision making component on the arc rather than the node. This modification together with the graph structure removes a further drawback of the classical decision tree – the problem of combinatorial explosion in the branching.

Well-defined rule navigation

Inference based systems are very sensitive to minor changes affecting the order in which rules are evaluated. This has the disadvantage of producing errors within the inferencing system, which may be difficult to uncover. One way to avoid this difficulty is to add some form of priority to the rules. However, this approach removes the advantage of the declarative nature of these systems and introduces brittleness akin to that in the classical decision tree systems. **Incanto's** graph has a natural priority order to the evaluation of the rule set. This is in plain view in the models and cannot be altered accidentally. Furthermore, the **Incanto** graph structure removes the possibility of cyclic rules errors.

Performance

The majority of systems using inference based processing are based on versions of the Rete algorithm developed by Forgy [For 82].

The latest version of Rete – Rete III, which is claimed to be “more than 300% faster than it’s competitors” [FI 05] has been tested by Fair Isaac to give “a real-time performance of more than 120 transactions a second per cpu (with more than 350 concurrent users per cpu and an average response time of 10 milliseconds)”[FI 05].

Incanto benchmarks at better than 1,000 transactions per second. There is no degradation within the engine with increasing number of users.

Benchmarks results of over 1,800 decisions a second with files exceeding 10GB in size have been observed in the **Incanto** batch engine.

Incanto achieves this performance by effectively sub-setting the rules being applied at any point. This is inherent in the graph.

Mitigation of brittleness

Non declarative decisioning methods are inherently prone to more brittleness than pure inferencing systems. In the case of classical decision trees this is due to the hard coded navigation within the binary structure. With inference based systems brittleness occurs with the introduction of priority in rules.

Incanto navigation is hard coded within the models. The brittleness introduced by this is mitigated by a number of factors. **Incanto** is graph based rather than tree based allowing processing paths to diverge and then converge. Strategies are constructed from multiple models each of which may be independently amended. Individual models are named and nodes allowing naming. Inter-model and required intra-model navigation can be name based rather than structure based. Additional arcs may be attached to a node or removed allowing modification to occur mid-model without the collapse of the structure.

Combinatorial explosion

Classical decision trees make binary decisions on the node. This generates exponentially increasing structure size with linearly increasing complexity. **Incanto** maintains linearly increasing model size with linearly increasing complexity. It achieves this in three ways. The first is to allow more than two arcs from any node. The second is to allow arbitrarily complex rules to be applied on any arc. The third is to allow processing paths to converge after diverging. The size of the model is thus kept in line with the complexity of the problem.

The Incanto fuzzy logic functionality

Incanto allows the combination of influences from multiple data items, through its high speed fuzzy logic functionality.

The **Incanto** method is based on the feedforward standard additive model (SAM) fuzzy system. This is a special case of additive fuzzy systems. [Kos 96]. The basis for these systems is the work by Takagi and Sugeno [Tak 85].

The Takagi-Sugeno fuzzy system

This method represents the conclusion of fuzzy inferencing by arithmetic functions.

It has the form:

IF x_1 is A_{r1} AND x_2 is A_{r2} AND x_3 is A_{r3} ... AND x_n is A_{rn} THEN $u = f_r(x_1, x_2, \dots, x_n)$

The function f_r with appropriate input values gives a direct mapping from the input space to the output space.

The connective operation is defined via the degree of relevance of the rule, the premise of the rule, and the function.

The final result is a weighted mean value of the connective operations over all the appropriate rules. Defuzzification is not required, as the crisp result value is directly determined by the inference operation.

The Takagi-Sugeno fuzzy system builds a combination of functions.

Fuzzy sets are often defined by multiple triangle fuzzifications. In older fuzzy systems the outline of the confined function surface was often used to determine the output fuzzy set.



In the limit this gives a flat straight line.

Kosko [Kos 94a] says: "To fix this I showed that you should add the two triangles to get the output set. I call this an additive fuzzy system".

Kosko [Kos 94a] also says: "I use the term FAM or fuzzy associative memory to describe how a fuzzy system works. Parallel and partially. Fire all rules to some degree".

Incanto follows the approach taken by Kosko with further modifications:

Incanto fires all of its fuzzy rules in parallel to derive the output fuzzy set. Selective inclusion of rule values are determined by alpha cut function. As in the Takagi-Sugeno fuzzy system and the standard additive model a singleton outcome is produced. These outcome values are normalised to 0-1.0 to allow use of them in further additive models.

Incanto's fuzzy logic functionality in summary:

Incanto uses linear and triangle fuzzification functions on the incoming data

Incanto uses the standard additive model

Outcomes are normalised

Defuzzification is performed by linear scaling of the outcome

Alpha cut allows selective inclusion of rule values

Models may be built with outcomes from previous models

In Conclusion

Incanto has been designed to overcome the performance problems associated with inference systems, and to mitigate the brittleness of standard decision tree implementations. The inclusion of fuzzy logic functionality adds the unique opportunity to make richer decisions, without impacting performance.

References

[FI 05] Rete III An Overview, Fair Isaac Executive Brief November 2005

[For 82] C. L. Forgy, Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem, Artificial Intelligence, 19 (1982), 17-37.

[Kos 94a] Kosko, B., Fuzzy Thinking, The New Science of Fuzzy Logic, Hyperion Books, 1994.

[Kos 94b] Kosko B.: Fuzzy systems as universal approximators. IEEE Trans. Computers 43, (11), 1324– 1333, 1994

[Kos 96] B. Kosko, "Fuzzy Engineering", Prentice Hall, 1996

[Tak 85] T. Takagi, and M. Sugeno, "Fuzzy identification of Systems and its Applications to Modeling and Control," IEEE Transactions on Systems, Man, and Cybernetics, 15(1), January-February 1985.

Written by Ryan Godridge, Qualia Systems Ltd

For more information, please contact

www.qualiasystems.co.uk

email: info@qualiasystems.co.uk