System Identification of Fuzzy Cartesian Granule Feature Models Using Genetic Programming

James F. BALDWIN, Trevor P. MARTIN, James G. SHANAHAN[1, 2]

Advanced Computing Research Centre, Dept. of Engineering Mathematics
University of Bristol, Bristol, BS8 1TR, ENGLAND
e-mail {Jim.Baldwin, Trevor.Martin, Jimi.Shanahan }@bristol.ac.uk

Abstract – A Cartesian granule feature is a multidimensional feature formed over the cross product of words drawn from the linguistic partitions of the constituent input features. Systems can be quite naturally described in terms of Cartesian granule features incorporated into additive models (if-then-rules with weighted antecedents) where each Cartesian granule feature focuses on modelling the interactions of a subset of input variables. This can often lead to models that reduce if not eliminate decomposition error, while enhancing the model's generalisation powers and transparency. Within a machine learning context the system identification of good, parsimonious additive Cartesian granule feature models is an exponential search problem. In this paper we present the G_DACG constructive induction algorithm as a means of automatically identifying additive Cartesian granule feature models from example data. G_DACG combines the powerful optimisation capabilities of genetic programming with a rather novel and cheap fitness function which relies on the semantic separation of concepts expressed in terms of Cartesian granule fuzzy sets in identifying these additive models. G_DACG helps avoid many of the problems of traditional approaches to system identification that arise from feature selection and feature abstraction such as local minima. G_DACG has been applied in the system identification of additive Cartesian granule feature models on a variety of artificial and real world problems. Here we present a sample of those results including those for the benchmark Pima Diabetes problem. A classification accuracy of 79.7% was achieved on this dataset outperforming previous bests of 78% (generally from black box modelling approaches such as neural nets and oblique decision trees).

1. Introduction

The ability to learn is considered the sine qua non of intelligence, which makes it an important concern for both cognitive psychology and artificial intelligence. The field of machine learning (ML), which crosses these disciplines, studies the computational processes that underlie learning in both humans and machines. The field's main objects of study are the artefacts; specifically algorithms that improve their performance with experience [33]. One of the primary goals of the field is to model the mechanisms that underlie human learning referred to as "cognitive simulation" in [44]. In achieving this we can find out how humans work and can perhaps help them to be better in their work.

[1] Supported by European Community Marie Curie Fellowship Program.

[2] Address correspondence to this author.

On the other hand from an engineering and system identification perspective, ever since the introduction of the first operational modern computer (Heath Robinson) in 1940 by Alan Turing's team, scientists and engineers have tried, with varying degrees of success, to increase its usefulness to mankind. Machine learning is one of the fields that can potentially make this more of a reality. Machine learning can be viewed as a means of automatically programming computers (or identifying systems) thus alleviating many of the problems facing our cyborg society:

> - It can help to model problem domains where domain knowledge is overly difficult to capture (too much, too little, too expensive etc...) for example in intelligent activities such as vision understanding [7], speech understanding [25]. Because machine learning can transform training data into knowledge it holds the potential of overcoming the knowledge acquisition bottleneck.
> - As data collection methods and storage technologies improve we face the challenge of a data flood. [18] note that "It has been estimated that the amount of data in the world doubles every 20 months ... earth observation satellites planned for the 1990s are expected to generate one terabyte (10^{15}) of data everyday". It is clearly infeasible for humans to trawl such data in search of patterns or relationships. This task falls within the field of Knowledge Discovery in Databases in which learning plays a key role in pan handling data for nuggets of knowledge (data mining).
> - To overcome the programming bottleneck (software lag) that results from deploying computers i.e. to automate the process of automation. This is seen as one of the most important areas of computer science over the next twenty years.

Within the field of ML there have been many fine successes and some serve to fuel other ML goals. For example, [13] presented an empirical study which "provides evidence that machine learning models can provide better classification accuracy than explicit knowledge acquisition". Kononenko [29] references 24 papers where inductive learning systems were actually applied in medical domains, such as oncology, liver pathology, prognosis of patient survival in hepatitis, urology, cardiology, gynaecology amongst many others. He remarks that "typically, automatically generated diagnostic rules slightly outperformed the diagnostic accuracy of physician specialists". In some cases the automatically programmed systems enhance human understanding. [36] provides further examples of success, with lots of fielded examples, in the field of machine learning.

Numerous approaches to system identification through machine learning exist. These can be quite easily categorised based upon two principles: performance accuracy and transparency (human understandability) of the induced model. To date most approaches that have focussed on the transparency of processes (through symbolic representation) have had only mild success in terms of performance accuracies compared to their mathematical counterparts. In this paper we present examples that support this, including a diabetes diagnosis system (see Section 5.2) where a symbolic learning approach such as ID3 [39] was applied to model this diagnosis process. Mathematically-derived approach such as neural networks were also applied, however the mathematically-based approaches outperform (in terms of accuracy) the symbolic approach even though the symbolic approach performs the best in terms of model transparency and understandability. With regard to current approaches to machine learning, these goals seem to be incompatible in that no one approach satisfies them both.

The work presented tries to fulfil both the desires of having accurate and understandable models that arise out of learning. This is enabled through the use of Cartesian granule features; multi-dimensional features built on words, thus enabling the paradigm "modelling with words". Systems can be quite naturally described in terms of Cartesian granule features incorporated into additive models (if-then-rules with weighted antecedents) where each Cartesian granule feature focuses on modelling the interactions of its constituent subset of input variables. Additive Cartesian granule feature models were originally introduced to overcome decomposition error, and also to enhance the model generalisation powers and transparency [6, 8, 41, 42].

In the context of automatically identifying additive Cartesian granule feature models from example data the discovery of good, highly discriminating, parsimonious Cartesian granule features, which adequately model the system at hand, is an exponential search problem. Numerous system identification algorithms exist (see Section 2 for a review), however most algorithms suffer from various problems that arise from poor feature selection and poor feature abstraction techniques. These problems include: inductive bias introduced by filter feature selection techniques; local optimum models that generally arise from the greedy nature of the search algorithms used in the identification process and also from treating feature selection and feature abstraction as two independent processes. Consequently, we propose the G_DACG constructive induction algorithm, which automatically identifies the important variable interactions and their abstractions that should be described using Cartesian granule features. The identified Cartesian granule features are then incorporated into additive models that generally provide good generalisation and transparency. G_DACG combines the powerful optimisation capabilities of genetic programming with a rather novel and cheap fitness function which relies on the semantic separation of concepts expressed in terms of Cartesian granule fuzzy sets in identifying these additive models. Furthermore it avoids some of the pitfalls of other identification algorithms such as local minima and provides a population-based (collective) approach to finding a solution as opposed to individual-based approaches.

The material in this paper is organised as follows: In Section 2 we overview system identification, focussing on the important roles feature selection and feature abstraction play in this process. Various structure identification strategies commonly used in machine learning are also reviewed. Section 3 serves as an introductory section to Cartesian granule features, a corresponding induction algorithm and additive models. In Section 4 we present the G_DACG constructive induction algorithm which automatically identifies additive Cartesian granule feature models. We illustrate this G_DACG algorithm on some problems in Section 5 and compare the results obtained with other standard machine learning approaches. Finally we finish off with some conclusions in Section 6.

2. System Identification through Induction

System identification through inductive learning can be viewed as the non-trivial general process of discovering useful models or knowledge about an application domain from observation data and background knowledge. System identification is a multi-faceted research area, drawing on methods, algorithms, and techniques from diverse umbrella fields such as knowledge representation, machine learning, pattern recognition, cognitive science, artificial intelligence, databases, statistics, probability, knowledge acquisition for expert systems and data visualisation. The unifying goal of these areas is the identification of predictive models from data and background knowledge that can simplify or enhance an application area. In this work, we are mainly concerned with the black box approach to system identification [34], in that we do not use any a priori knowledge in the model construction i.e. the model is constructed directly from the data or observations provided. Although expert or a priori knowledge in various guises can be incorporated into the system identification process, this is not addressed in this paper. Traditional approaches to systems modelling divide the problem of system identification into two sub-problems: those of structure identification and parameter identification.

2.1 Structure Identification

Structure identification is mainly concerned with selecting the language (i.e. the variables and their representations) in terms of which the models will be expressed. This language is defined in terms of the input features (and their derivations) and also for some forms of knowledge representations, in terms of the feature universe abstractions (sometimes linguistic). Feature selection and discovery form integral steps in this process. In fuzzy and other distribution based approaches (such as probability density estimation, radial basis function networks, etc.) a further level of identification is required where the granularity of the input feature universes needs to be determined. When dealing with prediction problems (i.e. output universe is continuous in nature) the granularity of the output universe will also have to be determined. These types of system are not considered here however, [42] gives details and examples of a heuristic approach to output granularity identification in the case of additive Cartesian granule feature modelling.

2.1.1 Feature selection and discovery

Feature selection can be viewed as the process of selecting those features that should be used in the subsequent steps of an induction or modelling process. Feature discovery can be viewed as a process of synthesising features from the base features and consequently involves feature selection. The synthesised features (and possibly the original feature set) can then be used by any induction process for the extraction of concept descriptions. Synthesised features tend to lead to more succinct and more discriminating concept descriptions. Numerous ways of synthesising new features have been proposed in the literature including [5, 11]; a genetic programming approach to the synthesis of compound features as algebraic expressions of base features. These synthesised features are subsequently used in fuzzy modelling. Several examples presented in [31, 32, 48] have incorporated feature synthesis indirectly into model construction through genetic programming. Logical rule induction systems such as

AQ17 [36] generate new features by combining base features using mathematical and logical operators in order to provide adequate concept descriptions. Feature synthesis and selection also forms an important part of neural network construction, where the hidden nodes may be viewed as higher order features that are discovered by the learning algorithm. Features are automatically selected as a result of training. Principal component analysis [23] offers an alternative route in constructing higher-order features from weighted combinations of base features based on variance measures. In the work presented here we construct Cartesian granule features based on the cross product of granules used to partition the base feature universes. In our work and in general one of the most critical steps in feature synthesis is the feature selection process.

There has been substantial work on feature selection in various fields such as pattern recognition, statistics, information theory, machine learning theory and computational learning theory. Numerous feature selection algorithms exist. [14, 28] characterise the various approaches as follows: those that "embed" the selection within the basic induction algorithm, those that use feature selection to "filter" features passed to induction, and those that treat feature selection as a "wrapper" around the induction process. Since feature selection plays a critical role in the discovery of Cartesian granule features we now briefly examine the various approaches to feature selection using these categories.

2.1.1.1 Embedded Approaches to Feature Selection

Embedded feature selection involves selecting features within the induction algorithm (single use/one-pass of induction process), where the general idea is to add or remove features from a concept description in response to an evaluation function e.g. prediction errors on unseen data. The various techniques differ mainly in the search strategies and heuristics used to guide the search. Because the search space can be exponentially large, managing the problem requires strong heuristics. For example, logical description induction techniques such as ID3, C4.5, and CART carry out a hill-climbing search strategy, guided by information-gain heuristics, to search programs (discover good features conjunctions), by working from general to specific. The ASMOD algorithm, which identifies B-spline and neuro-fuzzy models, and its various extensions [15, 24] are examples of an embedded feature selection strategy where the model is iteratively refined by modifying, adding or removing features. MARS [19], a identification algorithm for truncated spline models, is also an example of an embedded feature selection strategy.

These embedded techniques, due to the search mechanisms employed, are very vulnerable to starting points, and local minima [14, 15, 24, 28]. These search techniques work well in domains where there is little interaction amongst the relevant features. However, the presence of attribute interactions, can cause significant problems for these techniques. Parity concepts constitute the most extreme example of this situation, but it also arises in other target concepts. Embedded selection methods that rely on greedy search cannot distinguish between relevant and irrelevant features early in the search. Although combining forward selection and backward elimination to concept construction may help to overcome this problem. A better alternative may be to rely on a more random search such as simulated annealing, or a more random and diverse search technique such as genetic algorithms or genetic programming.

2.1.1.2 Filter Approaches to Feature Selection

A second general approach to feature selection introduces a separate process for this purpose that occurs before the basic induction step. For this reason [28] have termed them filter methods; they filter out irrelevant features before induction occurs. The pre-processing step generally relies on general characteristics of the training set to select some features and exclude others. Thus filtering methods are independent of the induction algorithm that will use their output and they can be combined with any such method. RELIEF [26] and FOCUS [1] and their extensions are amongst the more commonly used approaches to feature selection and have been shown to contribute significant improvements to a variety of induction approaches such as decision trees, nearest neighbours and naïve Bayesian classifiers [14]. RELIEF samples training instances randomly, summing a measure of the relevance of a particular attribute across each of the training instances. The relevance measure used is based upon the difference between the selected instance and *k* nearest instances of the same class and *k* nearest instances in the other classes ("near-hit" and "near-miss") [30]. REIGN [12] relies on the use of a feed forward neural networks (using back propagation learning algorithm) combined with a hill climbing search strategy to determine the features set that should subsequently be used by a fuzzy induction algorithm. Principal component analysis [23] is a form of filter that constructs higher-order features, orders them and selects the best such features. These features are then passed on to the induction algorithm. Filter approaches, while interesting and useful, totally ignore the demands and capabilities of the induction algorithm and thus can introduce an entirely different inductive bias to that of the induction algorithm [28]. This leads to the argument that the induction method planned for use with the selected features should provide better estimate of accuracy than a separate measure that has an entirely different inductive bias; this leads to the wrapper technique for feature selection.

2.1.1.3 Wrapper Approaches to Feature Selection

A third generic approach for feature selection is done outside the induction method but uses the induction method as the evaluation function. For this reason [28] refer to these as wrapper approaches. The typical wrapper approach conducts a search in the space of possible parameters. Each state in the parameter space corresponds to a feature subset and various other information depending on the induction algorithm used (for example the granularity of feature universe in the case of Cartesian granule features). Each state is evaluated by running the induction algorithm on the training data and using the estimated accuracy of the resulting model as a metric (other measures can also be used). Typical search techniques use a stepwise approach of adding or deleting features to previous states beginning with a state where all features or no features are present. The G_DACG constructive induction algorithm presented subsequently in Section 4.3 is an example of a wrapper approach to feature selection. The wrapper scheme has a long history within the statistics and pattern recognition communities [17, 22]. The major disadvantage of wrapper methods over filter schemes is the former's computational cost, which results from calling the induction algorithm for each parameter set evaluated. The approach is also susceptible to local minima when used in conjunction with stepwise search strategies.

2.1.2 Feature Abstraction

In the case of some forms of knowledge representation, an extra step in language selection is required; that of feature abstraction. Feature abstraction occurs usually in the form of partitioning. This helps reduce information complexity and in some cases enhances transparency and understandability. In fuzzy set based approaches to learning such as described in [46, 49] fuzzy partitioning is used. The granularity of the partitions in these approaches is determined heuristically. In the case of [49] granularity is determined using a clustering approach. In logical description induction techniques such as ID3, C4.5, and CART feature abstraction is achieved through crisp partitioning of the feature universes. This partitioning is normally accomplished by information-gain or purity heuristics. In general for these fuzzy set and decision tree based approaches the system identification algorithms perform the steps of feature selection and feature abstraction independently of each other. This can lead to models which are sub-optimum in nature. In the case of feedforward neural networks [21] partitioning is achieved through non-linear weighted sum combinations of features. The number of hidden nodes plays an important role in this type of partitioning and generally is determined either manually or automatically through network constructor algorithms [21]. In the case of Cartesian granule features, feature universes are abstracted by words that are characterised by fuzzy sets (linguistic universes). The level of granulation can be determined by expert input or automatically by the G_DACG constructive induction algorithm. G_DACG combines the feature selection and abstraction steps thus alleviating local minima problems. Characterising the granules by fuzzy sets provides the added advantage of smooth continuous behaviour across the universe of discourse. This is contrasted with a less desirable highly non-linear behaviour that typically results from crisp partitioning.

2.2 Parameter Identification

Parameter identification on the other hand can be viewed primarily as an optimisation procedure that fine-tunes the model language. In the case of polynomial curve fitting parameter identification consists of identifying the co-efficients in the polynomial. This is normally achieved by minimising the square of the output error. In most fuzzy set based systems parameter identification corresponds to identifying the location of the fuzzy sets that linguistically partition the variable universes [46, 49]. Once again commonly used procedures such as the mountain method [49] achieve parameter identification by minimising the output error using a back propagation type learning algorithm. In the case of additive Cartesian granule feature modelling parameter identification is concerned with selecting suitable granule characterisations and with setting up the class aggregation rules for the constituent Cartesian granule features: estimating the weights associated with the individual Cartesian granule feature (submodels); and tuning the rule filters. This is achieved by minimising the square of the output error.

3. Additive Cartesian Granule Feature Modelling

Cartesian granule features were originally introduced to overcome decomposition error, a problem which has plagued traditional AI and fuzzy approaches to knowledge based systems, and also to provide the transparency of traditional symbolic AI approaches [6, 8, 41, 42]. Cartesian granule features are a new type of multidimensional feature defined over the Cartesian product of words drawn from the linguistic partitions of the constituent feature universes. Variables defined over Cartesian granule universes can be viewed as multidimensional linguistic variables whose states are Cartesian granules i.e. Cartesian words where each word is characterised by a fuzzy set defined over the corresponding base variable universe.

3.1 Cartesian Granule Features

Here we give a brief overview of Cartesian granule features. A *granule* [50, 51], is a fuzzy set of points, which are labelled by a word. This collection of points is drawn together as result of indistinguishability, similarity, proximity or functionality. A *Cartesian Granule*, is an expression of form $W_1 \times W_2 \times .....\times W_m$ where each W_i is a word or label associated with a fuzzy set defined over the universe Ω_i and where "x" denotes the Cartesian product. A Cartesian granule can be visualised as a clump of elements in an n-dimensional universe sharing similar properties. A *Cartesian granule universe* is a discrete universe defined over $P_1\times P_2 \times .....\times P_m$ where each P_i is a linguistic partition of universe Ω_i and where "x" denotes the Cartesian product. In other words given a set of single attribute features $\{F_1, F_2...F_m\}$ defined over $\Omega_1 \times \Omega_2 \times .... \times \Omega_m$ where Ω_I is a universe of discourse over which F_i is defined, we form a linguistic partition P_i over each universe Ω_I. Partition P_i will consist of labelled fuzzy sets as follows :

$$\{A_{i1}, A_{i2}, ..., A_{ic}\}$$

We form the Cartesian granule space $\Omega_{P_1 \times P_2 \times ..... \times P_m}$ by taking the cross product of the words associated each fuzzy set across each partition P_i resulting in a discrete universe

$$\Omega_{P_1 \times P_2 \times ..... \times P_m} : \{ A_{11}A_{21}...A_{m1},\ A_{12}A_{22}...A_{m2},\ ....,\ A_{1c}A_{2c}...A_{mc}\}$$

where each Cartesian granule is merely a string concatenation of the individual fuzzy set labels A_{ij}.

A *Cartesian Granule Feature* is a feature defined over a *Cartesian Granule Space.* A *Cartesian granule fuzzy set* is a discrete fuzzy set defined over a *Cartesian granule universe.* Each *Cartesian granule* is associated with a membership value, which is calculated by combining the membership values, individual feature values have in the fuzzy sets which characterise the granules. For example, the Cartesian granule $w_{11} \times w_{21} \times ..... \times w_{m1}$ where each w_{i1} is the word associated with the first fuzzy subset in each linguistic partition P_i. Here the membership value associated with the Cartesian granule $w_{11} \times w_{21} \times ..... \times w_{m1}$ is calculated as follows:

$$m_{w_{11}}(x_1) \wedge m_{w_{21}}(x_2) .... \wedge m_{w_{m1}}(x_m)$$

where x_i is the feature value associated with the *i*-th feature within the data vector. Here the aggregation operator $\wedge$ can be interpreted as any T-norm [27, 40] such as product or min. The choice of conjunction operator is considered in [42].

3.1.1 A Cartesian Granule Fuzzy Set Example

The following example illustrates how to form a two dimensional Cartesian granule fuzzy set corresponding to a data vector. Using the single attributes *position* and *size* (attributes associated with objects in a digital image domain) we form a Cartesian granule universe. This is achieved by linguistically partitioning each of the base variable universes. One possible linguistic partition could be:

$$P_{position} = \{\text{left, middle, right}\} \quad \text{and} \quad P_{size} = \{\text{small, medium, large}\}.$$

This is depicted in Figure 1. Next we form the Cartesian granule universe defined over the words associated with the linguistic partitions. Our Cartesian granule space will consist of the following discrete elements:

$\Omega_{\text{position} \times \text{size}}$: *{ left.small, left.medium, left.large, middle.small, middle.medium, middle.large, right.small, right.medium, right.large}.*

If we define the position and size universes to be *[0, 100]* and *[0, 100]* respectively then the definitions of the fuzzy sets in partitions $P_{position}$ and P_{size} (in Fril notation [5])[3] could be:

left:[0:1, 50:0]
middle:[0:0, 50:1, 100:0]
right:[50:0, 100:1]

and

small:[0:1, 50:0]
medium:[0:0, 50:1, 100:0]
large:[50:0, 100:1].

Then taking a sample data tuple (in the form *<position, size>*) *<60, 80>* yields two fuzzy sets *{middle/.8+ right/.2}* and *{medium/.4+ large/.6}*. Next we form the Cartesian product of these fuzzy data to yield a fuzzy set in Cartesian granule space:

{middle.medium/.32 + middle.large/.48 + right.medium/.08 + right.large/.12}.

Here we have interpreted the combination operator $\wedge$ as product.

[3] A fuzzy set definition in Fril such as *middle*:[0:0, 50:1, 100:0] can be rewritten mathematically as follows (denoting the membership value of *x* in the fuzzy set *middle*):

$$m_{middle}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \dfrac{x}{50} & \text{if } 0 < x \leq 50 \\ \dfrac{100 - x}{50} & \text{if } 50 < x \leq 100 \\ 1 & \text{if } x \geq 100 \end{cases}$$

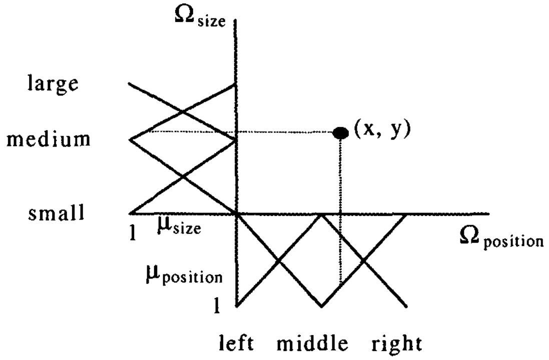


Figure 1 Fuzzy partitions of universes Ω_{size} and $\Omega_{positiion}$.

3.1.2 Cartesian Granule Fuzzy Sets Induction Algorithm

Notions fundamental to the formation of Cartesian granule features and fuzzy sets were presented in the previous sections. Here we extend these basic notions and show how they can be applied in a machine learning context. We present an induction algorithm that extracts concepts from example data in terms of Cartesian granule fuzzy sets.

Our proposed learning algorithm falls into the category of supervised learning algorithms. Within this framework databases of examples of the form:

$$< \vec{i}, \text{output} >$$

are utilised (for both training and testing) where $\vec{i}$ is a vector of values (where each value can be numeric or linguistic i.e. a single value, an interval value, or a fuzzy value) defined over the input attributes and are used to predict the output attribute value (which may be a single value, an interval value, or a fuzzy value). More formally a database D is defined over a set of attribute features $\{F_1, F_2 \ldots F_m\}$ defined in turn over the universes $\Omega_1, \Omega_2, \ldots\ldots., \Omega_m$. Here we have extended the notion of a conventional database attribute value to the case where uncertain or vague information can specified in terms of fuzzy subsets or interval values. Supervised learning algorithms normally address two types of problems, namely classification problems and prediction/regression problems. We present the induction algorithm from a classification problem perspective. A similar approach can be followed for prediction problems; instead of using the natural data partitioning provided by the output classification feature to build cartesian granule fuzzy sets corresponding to each class value, we generate a fuzzy partition in the output space (continuous) [42] and build Cartesian granule fuzzy sets corresponding to each concept (fuzzy set) in the output space.

3.1.2.1 Initialisation

We begin the whole induction process by selecting which features should be combined into Cartesian granule features. On this front we have proposed an automatic, near optimal, feature discovery algorithm based upon a genetic search, G_DACG, which will be presented in Section 4. However for now we can assume we will combine all the available input features into a Cartesian granule feature. Subsequently we form linguistic partitions over all attribute universes (both continuous and discrete) in the input space. The feature discovery algorithm will also determine automatically the granularity of the base feature universes and the granule characterisations. For the purposes of presenting this algorithm we assume that we have an expert who can indicate good linguistic partitions of the base feature universes. Having generated linguistic partitions over the universes of the selected features, we form the Cartesian granule universe Ω_{CG}. Next we split the database of examples into 2 parts namely the training database D_{train} and the testing database D_{test}. Subsequently we partition the database D_{Train} using the output classification values.

3.1.2.2 Extraction of Cartesian Granule Fuzzy Sets from Example Data

We extract a fuzzy set defined over the Cartesian word universe from example data, corresponding to each class in the output space. We begin by initialising a frequency distribution $DIST_{CG}$ defined over all the Cartesian granules in Ω_{CG}. We then take each training tuple for a class T_{ci} and construct the corresponding Cartesian granule fuzzy set (i.e. linguistic description of the data vector) CGF_{Ci} using the approach outlined in Section 3.1.1. Subsequently we form the least prejudiced distribution LPD_{Ci} [2, 5] corresponding to this fuzzy set CGF_{Ci} via its mass assignment. Next we update the overall frequency distribution $DIST_{CG}$ with this least prejudiced distribution LPD_{Ci}. We repeat this process for all training tuples in this class C_C. This results in frequency distribution $DIST_{CG}$ defined over the Cartesian granules corresponding to the class C_C. We take this distribution to correspond the least prejudiced distribution LPD_{CG}. We can then form a mass assignment corresponding to LPD_{CG}. Using the assumption of the least-prejudiced distribution we distribute probability masses uniformly within focal elements of the mass assignment and solve to find the associated Cartesian Granule Fuzzy Set. We repeat the above steps for each output classification C_C thereby extracting the corresponding class Cartesian granule fuzzy sets. These induced Cartesian Granule fuzzy sets can then be utilised to solve both classification and regression problems by incorporating them in to Fril product or evidential rules [4]. The induction algorithm for prediction problems is described in [8].

3.1.3 Additive Cartesian Granule Feature Models

[9, 42] highlighted the need for discovering structural decomposition of input spaces in order to generate Cartesian granule feature models that provide good generalisation and knowledge transparency. Cartesian granule features incorporated into evidential logic rule structures [4] provide a natural mechanism for capturing this type of decomposed approach to systems modelling [42] and is referred to as an additive model. The use of additive Cartesian granule feature models can lead to greatly simplified models which are comptractable (computationally tractable) and are amenable to human inspection, thus providing insight to the system being modelled, while also enhancing model generalisation.

The evidential logic rule structure [4] captures very naturally additive Cartesian granule feature models. Here classification problems are presented (see [8, 42] for prediction problems). A sample evidential logic rule structure is depicted in Figure 2. Here *CLASS* can be viewed as a fuzzy set consisting of a single crisp value (in the case of classification type problems). Each rule characterises the relationship between input and output data for a particular region of the output space i.e. a concept. An equivalence rule is normally used i.e. the support intervals associated with the rule are ((1 1)(0 0)). In the case of classification problem domains a rule is generated for each class in the output space.

The body of each rule consists of information expressed in terms of a list of problem domain features. Here each F_i represents a feature, which is either a single attribute feature, or Cartesian granule feature or some other type of derived feature. The values F_{iCLASS} of these features will typically be fuzzy sets defined over corresponding universe Ω_i (again it can be a fuzzy set defined over a single attribute universe or Cartesian granule universe and so on) corresponding to the output variable value *CLASS*. Notice how naturally we can treat features of heterogeneous forms in a very homogeneous manner using these representations.

((classification is *CLASS*) /* if */	*Head/Consequent*
(evlog FILTER (F_1 is f_{1CLASS}) w_1 : (F_i is f_{iCLASS}) w_i : (F_m is f_{mCLASS}) w_m	*Body/Antecedents and associated weights*
)):((1 1)(0 0))	*Rule Supports*

Figure 2 Fril evidential logic rule structure.

The weight term w_i associated with each body term in the evidential logic rule indicates its contributing weight of importance to this rule's conclusion. In generating evidential logic rules we need the additional step of calculating the weights associated with each body term. Since the values of each the body terms are fuzzy sets, regardless of the feature type being flat or Cartesian granule in nature, the weights can be estimated by measuring the semantic separation of the inter class fuzzy sets using semantic discrimination analysis (as presented in Section 4.2). Each rule body is associated with a filter or linguistic quantifier (expressed as a fuzzy set) that lends a linguistic interpretation to the support value generated by the rule body. This filter can be determined automatically from example data as discussed in Section 4.4.

3.1.4 Inference and Decision Making

Here we consider the general **inference** and **decision-making** processes used within this framework of knowledge representation for the classification problem domain - discrete output variable - (see [8, 42] for the prediction problem domain). As described in detail in the previous section, each rule consists of a body of features and their corresponding fuzzy set values. These features may be flat or Cartesian granule in nature. In the case of Cartesian granule features, when performing inference we require the additional inference step that interprets the input data vector $\vec{x}$ linguistically (see Section 3.1.1) which results in the Cartesian granule fuzzy set description *CGD* of $\vec{x}$. Then we merely carry out the semantic unification (SU i.e. the fuzzy set match via mass assignment theory [5]) between the class fuzzy set *CGF* and the data fuzzy set *CGD*. In otherwords, in the case of Cartesian granule features

$$\mathrm{SU(CGF} \mid \vec{\mathrm{x}}) = \mathrm{SU(CGF \mid CGD)}$$

where $\vec{x}$ corresponds to the input data and *CGD* to the Cartesian granule fuzzy set description of $\vec{x}$.

In general, when dealing with systems where the individual universes are granulated into fuzzy sets, multiple fuzzy sets and hence multiple fuzzy rules are called upon to deduce an answer from a particular case. For any particular test case, each rule is processed separately and then individual solutions are combined to give a final overall outcome. For each class rule in the rule set we calculate its respective level of support for the body and head of the rule. For evidential logic rules we calculate the body support B as

$$\mathrm{B}_{\mathrm{Class}} = \sum_{i=1}^{m} \mathrm{SU}(f_{iClass} \mid \vec{\mathrm{x}}_i)\mathrm{w}_{\mathrm{iClass}}$$

where $\mathrm{w}_{\mathrm{iCLASS}}$ is the weight of importance associated with feature i for class *Class*. Since we are utilising equivalence rules the support for the head clause of each class rule is equivalent to the support for the body of that rule [5].

Having calculated the level of support for each hypothesis *(classification is CLASS)*, some decision-making needs to take place. In the case of classification problems when the rule base is presented with an unclassified vector of data, inference is performed as described previously, thus yielding a point support value S_i for the hypothesis of the form *(classification is CLASS*$_i$*)* associated with each class rule R_i. Then the classification of the input data vector is determined as the class $CLASS_{\max}$ associated with the hypothesis with the highest support.

4. System identification of Additive Cartesian Granule Feature Models using G_DACG

Having described parsimonious additive model structure in terms of Cartesian granule features as a potentially effective means of representing models that provide good generalisation and model transparency, and having identified their construction as a feature selection and discovery process, here we present the G_DACG constructive induction algorithm which automates the process of additive Cartesian granule feature model discovery and construction. Genetic programming [31, 32] forms an integral part of the G_DACG feature discovery algorithm. Before describing the G_DACG algorithm we present the chromosome structure and fitness function used.

4.1 Chromosome Structure

There are infinite ways of forming the membership value associated with a Cartesian granule in a Cartesian granule fuzzy set [8, 41]. This would correspond to an infinite function set in genetic programming terms. To date we have mainly used two operators, product and min operators. Both the product and min are intuitive conjunction operators [42]. However empirical evidence on various problem domains seems to suggest that there is very little difference between the effectiveness of both these operators [8, 41]. Consequently we have reduced our function set to the product operator *CGProduct*. At a later date it is hoped to allow a richer function set and genetically select appropriate conjunction operators. The arity of the *CGProduct* function can vary from one to the number of available base features, though parsimonious (low dimensional) Cartesian granule features are encouraged. This desire/behaviour is encoded in the fitness function.

Our terminal set consists of all the base features we wish to use in systems modelling along with their respective granularity range (abstraction). For example if we have 2 base features *f1* and *f2* and we allow a granularity range of [2..4] for each base feature, then, we would have a terminal set made up of the following:

{f1_G2, f1_G3, f1_G4, f2_G2, f2_G3, f2_G4}

where $f_i\_G_j$ corresponds to base feature *i* and with a granularity of *j*.

Since we are currently dealing with just one function, *CGProduct*, we can reduce the complexity of our chromosome structure from a tree structure to a list structure. This becomes feasible as a result of the discrete nature of Cartesian granule features. The granularity range for the base feature universes is very much feature and problem dependent, although a range of [2..15] is thought to be sufficient for most problem domains. The distribution of fuzzy sets across each of the feature universes is set, by default, to uniform, in order to decrease the search complexity. However, this could be automatically determined using the genetic search approach.

4.2 Fitness

The most important and difficult concept of genetic programming is the determination of the fitness function. The fitness function dictates how well a discovered program is able to solve the problem. The output of the fitness function is used as the basis for selecting which individuals get to procreate and contribute their genetic material to the next generation. The structure of the fitness function will vary greatly from problem to problem. In the case of Cartesian granule feature identification the fitness function needs to find Cartesian granule features which give good class separation (class corresponds to specific areas of the output variable universe) and are parsimonious. Consequently when used in fuzzy modelling these features should yield high classification accuracy with low computational overhead along with transparent reasoning. Cartesian granule features can be determined individually for each class in the problem domain (heterogeneous feature discovery) or alternatively in unison (homogeneous feature discovery). The fitness for an individual Cartesian granule feature (for a particular class or all classes) is a weighted combination of the discrimination (separation) of the individual and the parsimony of the individual, which is measured in terms of dimensionality of the individual and the size (cardinality) of the individual's universe of discourse. In order to calculate the semantic discrimination of a

Cartesian granule feature we need to construct the Cartesian granule fuzzy sets corresponding to each class in the output universe. Subsequently the process of semantic discrimination analysis determines the mutual dissimilarity of individuals, measured in terms of the point semantic unifications between the Cartesian granule fuzzy set corresponding to the current class CGF_i and the other class CG fuzzy sets CGF_j. This is written more succinctly as follows:

$$\text{Discrimination}_i = 1 - \underset{\substack{j=1 \\ j \neq i}}{\overset{C}{\text{Max}}} \Pr(\text{CGF}_i \mid \text{CGF}_j)$$

where C corresponds to the number of classes in the current system.

The dimensionality factor corresponds to the number of base features making up a Cartesian granule feature. The size (cardinality) of a Cartesian granule feature universe is simply the number of Cartesian granules in the corresponding universe. During the process of evolution it is important to promote individuals that have high discrimination, low dimensionality and small universe size. The latter of these two desires are expressed linguistically using the fuzzy sets depicted in Figure 3.

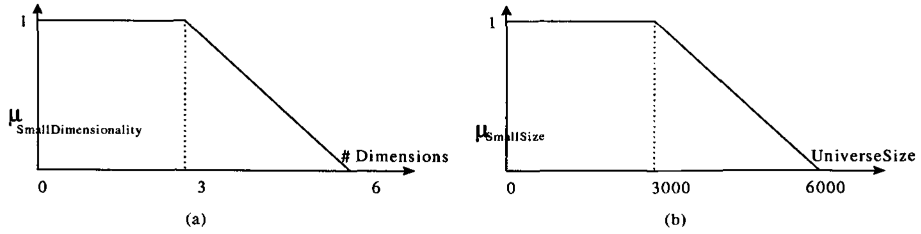


Figure 3 (a) Fuzzy set corresponding to small dimensionality in Cartesian granule features. (b) Fuzzy set corresponding to small size of Cartesian granule feature universes.

We combine the individual factors in the following manner:

$$\begin{aligned}\text{Fitness}_i = {} & W_{\text{Dis}} * \text{Discrimination}_i + W_{\text{Dim}} * \mu_{\text{SmallDim}}(\text{Dimensionality}_i) + \\ & W_{\text{USize}} * \mu_{\text{SmallUinv}}(\text{UniverseSize}_i)\end{aligned}$$

where W_{Dis}, W_{Dim} and W_{USize} take values in the range [0..1] and sum to 1. Since Cartesian granule features of high discrimination are desirable regardless of other criteria W_{Dis} tends to take values in the range [0.7..0.8]. The remaining weight is split evenly amongst W_{Dim} and W_{USize}. The weights are determined heuristically from trial runs.

4.3 Genetic Discovery of Additive Cartesian Granule Feature Models (G_DACG)

The discovery of good, highly discriminating, parsimonious Cartesian granule features is an exponential search problem that forms one of the most critical and challenging tasks in the additive model identification. Obviously no parameter optimisation algorithm can overcome shortcomings in structure identification. An additive model composed of Cartesian granule features that are too simple or too inflexible to represent the data will have a large bias, while one which has too much flexibility (i.e. redundant structure) may fit idiosyncrasies found in the training set producing models that generalise poorly; in this case the model's variance is too high. This is an example of the classical bias/variance dilemma presented in [20]. Bias and variance are complementary quantities, and the best generalisation is obtained when we have the best compromise between the conflicting requirements of small bias and small variance.

In order to find the optimum balance between bias and variance we need to have a way of controlling the effective complexity of the model. This trade-off is incorporated directly into the G_DACG discovery algorithm at two levels; one in terms of a fitness function for the individual Cartesian granule features (submodel level) and the other at aggregate model level where lowly significant features based on semantic discrimination analysis are eliminated. In the case of additive Cartesian granule features models, both the bias and variance can be drawn towards their minimum, by adding, removing, or altering (granularities, granule characterisations) the constituent Cartesian granule features, thereby generating models which tend to generalise better and have a simpler model structure; i.e. Occam's razor, where all things being equal the simplest is most likely to be the best.

As was seen earlier in Section 2.1, the search algorithm plays a big part in the discovery of good features. It can influence what parts of the parameter space are or are not evaluated due to local minima, starting states and computational constraints. Each state in the parameter space corresponds to a feature subset and the granularity of the individual base features i.e. the feature selection and feature abstraction steps are combined. The size of the finite space of all possible Cartesian granule features for any problem given a finite number of base features is given by the following equation:

$$\sum_{granularity=\min Gran}^{\max Gran} \sum_{\dim=1}^{MaxDim} \mathrm{NumOfFeatu\ res}\, C_{\dim} * (granularit\ y)^{\dim}$$

Note that in this case, the granule characterisations are assumed to be fixed (for example triangular fuzzy sets), otherwise, the complexity could potentially increase by another order of magnitude. For a sample problem, like the Pima Indian diabetes problem presented later in Section 5.2, the number of possible Cartesian granule features runs into millions if the eight base features are considered with base feature granularity ranges of [2, 15]. In general the search space will be of the order of millions, increasing exponentially with the permitted Cartesian granule features dimensionality. Consequently, traditional approaches to feature discovery would prove computationally intractable even for low-dimensional problems. Here we propose an additive Cartesian granule feature model constructive induction algorithm centred around a pseudo-random, distributed search paradigm based upon natural selection and population genetics; genetic programming. The genetic search paradigm, due to its distributed nature, avoids pitfalls such as local minima by exploring large areas of the search space in parallel. Currently we use the steady state flavour of genetic programming (SSGP) [31, 47]. SSGP permits overlapping generations and when used in conjunction with k-tournament selection avoids the problem of losing good individuals. We use a flavour of

SSGP where duplicate children are discarded rather than inserted into the population [47]. This helps promote diversity and avoids premature convergence in the population. Furthermore since the individuals will solve problems collectively (rather than individually), in the case of additive Cartesian granule feature modelling, this flavour of genetic programming is deemed to be appropriate. From a feature selection point of view, the G_DACG algorithm could be classified as wrapper feature selection algorithm in that it uses the Cartesian granule feature induction algorithm to evaluate the relevance of the individual Cartesian granule features.

The key steps involved in the G_DACG algorithm are as follows (see Figure 4 for a schematic):

- Generate a random set of individual Cartesian granule features
- Assign a fitness value to each individual
- REPEAT
    - Generate *n* new fitnessed children
    - Insert new children into population
    - Eliminate *n* individuals from the population
    - Determine best Additive Model
- UNTIL a satisfactory solution or the number of generations expires.
- Determine best Additive Model

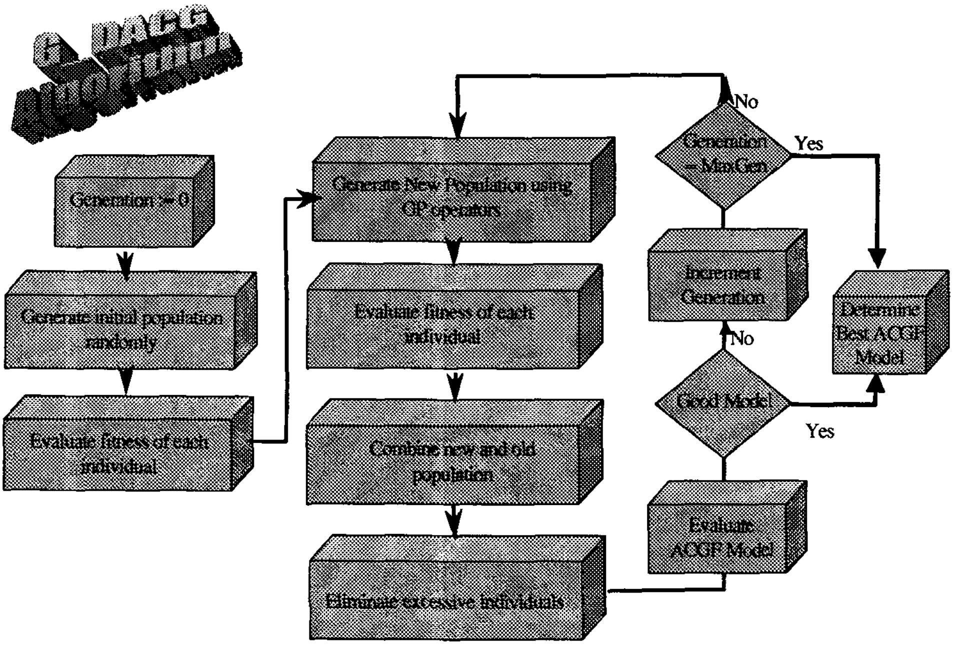

G_DACG Algorithm
Generation := 0
Generate initial population randomly
Evaluate fitness of each individual
Generate New Population using GP operators
Evaluate fitness of each individual
Combine new and old population
Eliminate excessive individuals
Evaluate ACGF Model
Good Model
No
Yes
Increment Generation
Generation = MaxGen
No
Yes
Determine Best ACGF Model

Figure 4: G_DACG constructive induction algorithm.

Determining the best additive model from the discovered Cartesian granule features can be performed at the end of each iteration of the genetic search or at the termination of the algorithm. This takes the form of selecting *n* of the best features (either heterogeneous or homogeneous discovered features) from the current population and constructing the corresponding additive model (i.e. determine the parameters of the model, see next section). Then superfluous Cartesian granule features are removed from the model by eliminating lowly contributing features, using a process known as backward elimination [17], thereby decreasing the additive model's bias and its variance. Alternatively, using the final population of individuals (or a subset), a genetic search can be performed of the possible additive models (of limited dimensionality). Structure identification is also concerned with the number of rules (and hence the number of classes in the output space) in our model. When dealing with classification type problems, structure identification of this type reduces to building one rule for each class. One technique that has been developed to speed up the evaluation process is to cache the fitnesses of the hypothesised Cartesian granule features. In genetic searches, while diversity tends to be relatively high, Cartesian granule features can be visited repeatedly. Exploiting the cached results can lead to significant computational gains.

4.4 Parameter Identification

Parameter identification is concerned primarily with setting up the class aggregation rules for the constituent Cartesian granule features: i.e. estimating the weights associated with the individual Cartesian granule feature (submodels) and tuning the class rule filters. We estimate the weights associated with each Cartesian granule feature using semantic discrimination analysis. Other optimisation techniques could be used. Since the submodels are being aggregated using the evidential logic rule another degree of parameter identification needs to be performed; that of learning the filter. This is addressed in [42] where a data driven optimisation algorithm centred on Powell's direction set minimisation technique is presented. An alternative parameter identification technique based upon the Mass Assignment Neuro Fuzzy (MANF) framework, where neural network learning algorithms can be applied to learn the submodel aggregation function, is also considered in [42].

5. Results

The C_DACG algorithm has been illustrated and compared with other machine learning approaches on a variety of problem domains including object recognition [42] and plant control [10]. Here illustrate the approach on some benchmark machine learning problems.

5.1 Ellipse Problem

The ellipse problem is a binary classification problem based upon artificially generated data from the universe $\Re x \Re$. Points satisfying an ellipse inequality are classified as legal while all other points are classified as illegal. This is graphically depicted in Figure 5 for the ellipse inequality

$$x^2 + 2y^2 \leq 1.$$

Thus there are two single attribute input features, *X* and *Y*. The universe of *X*, Ω_X is taken to be [-1.5, 1.5] and similarly the universe of *Y*, Ω_Y is taken to be [-1.5, 1.5]. Different training, control (validation) and test datasets, comprising of 1000, 300 and 1000 data vectors respectively, were generated using a pseudo-random number stream.

An equal number of data samples for each class were generated. Each data sample consists of a triple <*X*, *Y*, *Class*>, where *Class* adopts the value *0* for *illegal* indicating that the point <*X*, *Y*> does not satisfy the ellipse inequality, and the value *1* for *legal* otherwise.

5.1.1 A G_DACG Run on the Ellipse Problem

Here we present the steps and parameter settings involved in a typical run of the G_DACG constructive induction algorithm; we construct an additive Cartesian granule feature model for the Ellipse problem. Genetic programming is integral part of the G_DACG algorithm genetically evolving Cartesian granule features. As a result a lot of the algorithm parameters are GP related. In a typical GP run the population size is limited to 20 chromosomes, due to the small nature of the problem. Initial populations are generated using the ramped-half-and-half procedure [31] i.e. half-random length chromosomes and half full-length chromosomes. The length of chromosome range, in the initial population and in subsequent generations is problem dependent but parsimony is promoted. The k-tournament selection parameter *k* was set to 3 for this problem. The G_DACG algorithm iterated for thirty generations (or if the stopping criterion was satisfied it halted earlier, arbitrarily set at 100% accuracy) and at the end of each generation three of the best Cartesian granule features were selected from the current population. The selected features were then used to form an additive Cartesian granule feature model – best of generation model. Backward elimination based on fitness was employed, eliminating extraneous lowly contributing features. Once the main part of the G_DACG algorithm finished three of the best features that were discovered during the G_DACG iterations were combined to form an ACGF model – overall best model. Again backward elimination based on fitness was employed. Subsequently the model with the highest accuracy was selected from the best of generation models and the overall best model as a suitable ACGF model for ellipse problem. In the case of this problem the best discovered ACGF model was generated by taking the three best Cartesian granule features from generation 10 of a G_DACG run. This yielded the rule-based model depicted in Figure 6. The rule corresponding the legal class consists of three Cartesian granule features, while the rule for the illegal case consists of just 2 features. Backward elimination based upon semantic discrimination eliminated the third feature from the illegal rule. The optimally determined filters correspond to the "true" filter for this model (not shown in Figure 6). The discovered additive model yields an accuracy of 98.7%. A trapezoidal fuzzy set with 60% overlap was determined to be the best granule characterisation in the case of the evaluated models.

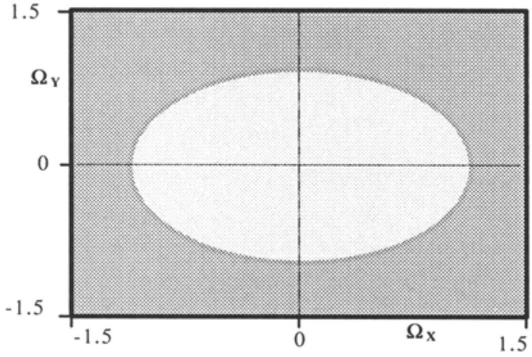


Figure 5: Ellipse inequality in Cartesian space. Points in lightly shaded region satisfy the ellipse inequality and thus are classified as legal. Points in darker region are classified as illegal.

5.1.2 Ellipse Results Comparison

Table 1 presents a summary of some of the best results achieved using various inductive learning approaches. All of the approaches examined here do very well in modelling the ellipse problem from a generalisation perspective. The discovered Cartesian granule features are very parsimonious in nature compared to the more complex two-dimensional Cartesian granule features model presented in Table 1. The granularity of the universes used in the additive models is much lower (three or four words) compared what is required in the non-additive model (11 words) in order to achieve the same level of accuracy. This reduction in granularity has been achieved by modelling the important decomposed variable interactions as opposed to focussing on the model of a single composed interaction.

```
((Predicted class for ellipse in case (CASE) is positive)
 (evlog POSITIVE_FILTER (
        (cgValue of ((X 4))) in (CASE) is positiveClass)0.2426
        (cgValue of ((Y 4))) in (CASE) is positiveClass) 0.367
       (cgValue of ((X 4)(Y 3))) in (CASE) is positiveClass) 0.39  ) ) ):((1 1)(0 0))

((Predicted class for ellipse in case (CASE) is negative)
 (evlog NEGATIVE_FILTER (
        (cgValue of ((Y 4))) in (CASE) is negativeClass) 0.396
       (cgValue of ((X 4)(Y 3))) in (CASE) is negativeClass) 0.604 ) ) ):((1 1)(0 0))
```

Figure 6: An example of an additive Cartesian feature model in Fril for the ellipse problem. This model gives over 98.7% accuracy on test cases.

Table 1: Summary of ellipse problem using various learning approaches.

Approach	Features details	% Accuracy
Additive Cartesian granule feature model	((X 4)) ((Y 4))((X 4) (Y 3)) - Legal ((Y 4))((X 4) (Y 3)) - Illegal	98.7
Two-dimensional Cartesian granule features	(X, Y), Granularity = 11, 60% Overlapping Trapezoids	98.8
Data browser(evidential logic rules)	X, Y(non-smoothed fuzzy sets)	94
Neural network	X, Y, and 3 hidden nodes	99.5
MATI	X, Y [3]	99

5.2 Modelling Pima Diabetes Detection Problem

The problem posed here is to predict whether a patient would test positive or negative for diabetes according to the World Health Organisation criteria given a number of physiological measurements and medical test results. The dataset was originally donated by Vincent Sigillito, Applied Physics Laboratory, John Hopkins University, Laurel, MD 20707 and was constructed by constrained selection from a larger database held by the National Institute of Diabetes and Digestive and Kidney Diseases [45]. It is publicly available from the machine learning repository at UCI [35]. All the patients represented in this dataset are females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona, USA. There are eight input attributes the values of which are used to predict the output classification of "testing positive for diabetes" and "testing negative for diabetes". These input-output attributes and their corresponding feature numbers (used for convenience) are listed in Table 2. This is a binary classification problem with a classification value of 1 corresponding to "testing positive for diabetes" and a value of 2 corresponding to "testing negative for diabetes". There are 500 examples of class 1 (positive) and 268 examples of class 2.

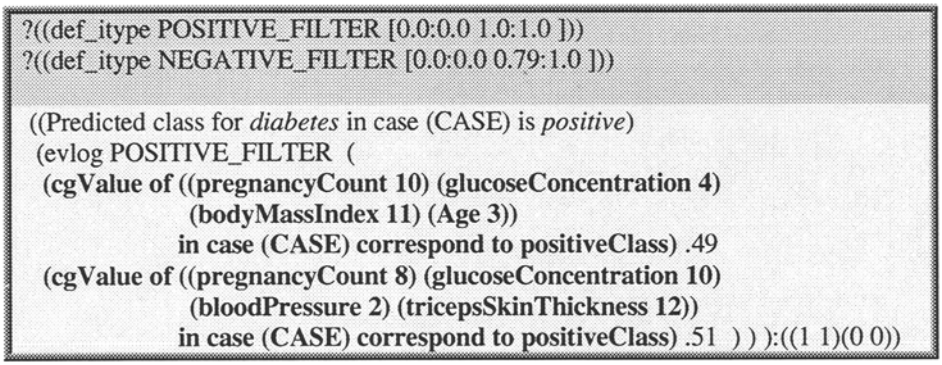
Table 2: Input base Features for the Pima Diabetes Problem.

No.	Class
0	Number of times pregnant
1	Plasma glucose concentration in an oral glucose tolerance test
2	Diastolic blood pressure(mm/Hg)
3	Triceps skin fold thickness(mm)
4	2-hour serum insulin (mu/U/ml)
5	Body mass index (kg/m^2)
6	Diabetes pedigree function
7	Age(years)

5.2.1 Additive Cartesian Granule Feature Modelling of Pima Diabetes Problem

The Pima diabetes data set of 768 tuples was split class-wise, approximately as follows: 60% of data allocated to training, 15% to validation and 25% to testing. We applied the

G_DACG constructive induction algorithm to the Pima diabetes problem. All eight base features were considered and Cartesian granule features of dimensionality up to five with granularity ranges of [2, 12] were considered (while parsimony was promoted in the form of the fitness function used) thus yielding a multi-million node search space. The k-tournament selection parameter *k* was set to 4 for this problem. The G_DACG algorithm iterated for thirty generations (or if the stopping criterion was satisfied it halted earlier, arbitrarily set at 90% accuracy) and at the end of each generation five of the best Cartesian granule features were selected from the current population. The selected features were then used to form an additive Cartesian granule feature model – best of generation model. Backward elimination based on fitness was employed, eliminating extraneous lowly contributing features. Once the main part of the G_DACG algorithm finished five of the best features that were discovered during the G_DACG iterations were combined to form an ACGF model – overall best model. Again backward elimination based on fitness was employed. Subsequently the model with the highest accuracy on test data was selected from the best of generation models and the overall best model as a suitable ACGF model for diabetes detection in Pima Indians. In the case of this problem the best discovered ACGF model was generated by taking the five best Cartesian granule features that were visited during the genetic search phase. During the genetic search process the granule characterisations were set to trapezoidal fuzzy sets with 50% overlap. However in this phase of the process, a variety of granule characterisations were investigated. A trapezoidal fuzzy set with 70% overlap was determined to be the best granule characterisation in the case of the evaluated models. The best discovered model from both a model accuracy and simplicity perspective consists of two Cartesian granule features (arrived at by backward elimination), yielding a model accuracy on test data of 79.7%. The Fril code corresponding to this model is presented in Figure 7. The negative class rule filter in this case is more disjunctive or optimistic in nature than its positive counterpart. This optimism may arise from the fact that a single feature may be adequate to model this class.



```
?((def_itype POSITIVE_FILTER [0.0:0.0 1.0:1.0 ]))
?((def_itype NEGATIVE_FILTER [0.0:0.0 0.79:1.0 ]))

 ((Predicted class for diabetes in case (CASE) is positive)
  (evlog POSITIVE_FILTER (
   (cgValue of ((pregnancyCount 10) (glucoseConcentration 4)
                (bodyMassIndex 11) (Age 3))
               in case (CASE) correspond to positiveClass) .49
   (cgValue of ((pregnancyCount 8) (glucoseConcentration 10)
                (bloodPressure 2) (tricepsSkinThickness 12))
               in case (CASE) correspond to positiveClass) .51 ) ) ):((1 1)(0 0))
```

Figure 7: An example of an additive Cartesian feature model in Fril for Pima diabetes detection. This model gives over 79.69% accuracy on test cases. Note only the positive rule is shown here. The negative rule has a similar structure.

5.2.2 Pima Diabetes Results Comparison

The Pima diabetes dataset serves as a benchmark problem in the field of machine learning and has been tested on many learning approaches. Table 3 compares some of the results of the more common machine learning techniques with the ACGF modelling approach. The Pima diabetes database illustrates a parity-problem-type/chaotic behaviour (i.e. change one input feature value and the classification also changes) especially when the data is projected onto lower dimensional feature spaces. This is reflected in the lack of semantic separation of concepts represented in lower dimensional Cartesian granule features. The discovered ACGF models support this in that they consist of submodels of high dimensionality.

Table 3: Comparison of results for the Pima diabetes detection problem.

Approach.	Accuracy(%)
Additive Cartesian granule feature Model	79.7
Mass Assignment based MATI [3]	79.7
Oblique Decision Trees [16]	78.5
Neural Net (normalised Data)	78
C4.5 [38]	73
Data browser	70

The Pima diabetes problem is a notoriously difficult machine learning problem. Part of this difficulty arises from the fact the dependent output variable is really a binarised form of another variable which itself is highly indicative of certain types of diabetes but does not have a one-to-one correspondence with the condition of being diabetic [37]. To date no machine learning approach has obtained an accuracy higher than 78% [35]. The discovered ACGF models have yielded very high accuracies (79.7%), outperforming other machine learning approaches (see Table 3).

6. Conclusions

The focus and motivation behind this work was the development of an automatic system identification process that leads to additive Cartesian granule feature models that are ultimately understandable not only by computers but also by experts in the domain of application and that perform effectively. This has resulted in the development of a new constructive induction algorithm – G_DACG. G_DACG avoids many of the pitfalls of other induction algorithms that arise from poor feature selection and abstraction. G_DACG was illustrated on variety of problems (synthetic and real world) and the discovered models in general performed as well or outperformed (in terms of accuracy) other well-known techniques in the field. From a model transparency perspective, the G_DACG algorithm, while yielding glassbox models in particular for the ellipse problem, needs further work when applied to real world problems. This is highlighted by the models discovered in the Pima diabetes problem where the Cartesian granule feature are of high dimensionality and consist of relatively high granularity. Cartesian granule features do however lay the foundations for a learning paradigm that provides the accuracy of mathematical approaches, while also achieving model transparency. Current work [43] is addressing the transparency issue as follows:

- Increase the expressiveness of the hypothesis language from attribute-value to relational.
- Hierarchical modelling (somewhat related to relational descriptions of concepts) is a promising approach that facilitates the capture of deep knowledge representation as opposed to the relatively shallow representations (considered here) and in most learning approaches.

References and Related Bibliography

1. **H. Almuallim and T. G. Dietterich** (1991), "*Learning with irrelevant features*", in Proc. AAAI-91, Anaheim, CA, pp 547-552.
2. **J. F. Baldwin** (1991) "*A Theory of Mass Assignments for Artificial Intelligence*", in IJCAI '91 Workshops on Fuzzy Logic and Fuzzy Control, Sydney, Australia, Lecture Notes in Artificial Intelligence, A. L. Ralescu, Editor 1991, pp. 22-34.
3. **J. F. Baldwin, J. Lawry and T.P. Martin** (1997), "*Mass assignment fuzzy ID3 with applications*", in Proc. Fuzzy Logic: Applications and Future Directions Workshop, London, UK, pp 278-294.
4. **J. F. Baldwin, T. P. Martin and B. W. Pilsworth** (1988) "*FRIL Manual*", FRIL Systems Ltd, Bristol, BS8 1QX, UK.
5. **J. F. Baldwin, T. P. Martin and B. W. Pilsworth** (1995) "*FRIL - Fuzzy and Evidential Reasoning in A.I.*", Research Studies Press(Wiley Inc.), ISBN 086380159 5.
6. **J. F. Baldwin, T. P. Martin and J. G. Shanahan** (1996) "*Modelling with Words using Cartesian Granule Features*", (Report No. ITRC 246), Advanced Computing Research Centre, Dept. of Engineering Maths, University of Bristol, UK.
7. **J. F. Baldwin, T.P. Martin and J. G. Shanahan** (1997), "*Fuzzy logic methods in vision recognition*", in Proc. Fuzzy Logic: Applications and Future Directions Workshop, London, UK, pp 300-316.
8. **J. F. Baldwin, T. P. Martin and J. G. Shanahan** (1997), "*Modelling with words using Cartesian granule features*", in Proc. FUZZ-IEEE, Barcelona, Spain, pp 1295-1300.
9. **J. F. Baldwin, T.P. Martin and J. G. Shanahan** (1998), "*Aggregation in Cartesian granule feature models*", in Proc. IPMU, Paris, pp 6.
10. **J. F. Baldwin, T. P. Martin and J. G. Shanahan** (1998) "*Controlling with words using automatically identified fuzzy Cartesian granule feature models*", (To Appear) International Journal of Approximate Reasoning - Special issue on Fuzzy Logic Control: Advances in Methodology, N/A, pp. 37.
11. **J. F. Baldwin and B. W. Pilsworth** (1997), "*Genetic Programming for Knowledge Extraction of Fuzzy Rules*", in Proc. Fuzzy Logic: Applications and Future Directions Workshop, London, UK, pp 238-251.
12. **A. Bastian** (1995) "*Modelling and Identifying Fuzzy Systems under varying User Knowledge*", PhD Thesis, Meiji University, Tokyo,
13. **A. Ben-Davis and J. Mandel** (1995) "*Classification accuracy: machine learning vs. explicit knowledge acquisition*", Machine Learning, **18**, pp. 109-114.
14. **A. L. Blum and P. Langley** (1997) "*Selection of relevant features and examples in machine learning*", Artificial Intelligence, **97**, pp. 245-271.

15. **K. M. Bossley** (1997) "*Neurofuzzy Modelling Approaches in System Identification*", PhD Thesis, Department of Electrical and Computer Science, Southampton University, UK,

16. **N. Cristianini** (1998) "*Application of oblique decision trees to Pima diabetes problem*", Personal Communication, Department of Engineering Mathematics, University of Bristol, UK.

17. **P. A. Devijer and J. Kittler** (1982) "*Pattern Recognition: A Statistical Approach*", Prentice-Hall, Englewood Cliffs, NJ.

18. **W. J. Frawley, G Piatetsky-Shapiro and C. J. Matheus** (1991) "*Knowledge Discovery in Databases: An Overview*", in Knowledge Discovery in Databases, G. Piatetsky-Shapiro and W. J. Frawley, Editors 1991, AAAI Press/MIT Press. Cambridge, Mass, USA. pp. 1-27.

19. **J. H. Friedman** (1991) "*Multivariate Adaptive Regression Splines*", The Annals of Statistics, **19**, pp. 1-141.

20. **S. Geman, E. Bienenstock and R Doursat** (1992) "*Neural networks and the bias/variance dilemma*", Neural computation, **4**, pp. 1-58.

21. **J. Hertz, K. Anders and R. G. Palmer** (1991) "*Introduction to the Theory of Neural Computation*", Addison-Wesley, New York.

22. **A. G. Ivanhnenko** (1971) "*Polynomial theory of complex systems*", IEEE Transactions on Systems, Man and Cybernetics, **1**(4), pp. 363-378.

23. **I. T. Jolliffe** (1986) "*Principal Component Analysis*", Springer, New York.

24. **T. Kalvi** (1993) "*ASMOD: an algorithm for Adaptive Spline Modelling of Observation Data*", International Journal of Control, **58**(4), pp. 947-968.

25. **M. Kay, J. M. Gawson and P. Norvig** (1994) "*Verbmobil: A translation system for face-to-face dialog*", CSLI Press, Stanford, California, USA.

26. **K. Kira and L Rendell** (1992), "*A practical approach to feature selection*", in Proc. 9th Conference in Machine Learning, Aberdeen, Scotland, pp 249-256.

27. **G. J. Klir and B. Yuan** (1995) "*Fuzzy Sets and Fuzzy Logic, Theory and Applications*", Prentice Hall, New Jersey.

28. **R Kohavi and G. H. John** (1997) "*Wrappers for feature selection*", Artificial Intelligence, **97**, pp. 273-324.

29. **I. Kononenko** (1993) "*Inductive and Bayesian learning in medical diagnosis*", Artificial Intelligence, 7, pp. 317-337.

30. **I. Kononenko and S J Hong** (1997) "*Attribute selection for modelling*", FGCS Special Issue in Data Mining, (Fall), pp. 34-55.

31. **J. R. Koza** (1992) "*Genetic Programming*", MIT Press, Massachusetts.

32. **J. R. Koza** (1994) "*Genetic Programming II*", MIT Press, Massachusetts.

33. **P. Langley** (1996) "*Elements of Machine Learning*", Morgan Kaufmann, San Francisco, CA, USA.

34. **L. Ljung** (1987) "*System identification: theory for the user*", Prentice Hall, Englewood Cliffs, New Jersey 07632.

35. **C. J. Merz and P. M. Murphy** (1996) "*UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA*", University of California, Irvine, CA.

36. **R. S. Michalski, I. Bratko and M. Kubat** (Ed), (1998), "*Machine Learning and Data Mining*", Wiley, New York.

37. **D. Michie, D. J. Spiegelhalter and C. C. Taylor** (1993) *"Dataset Descriptions and Results"*, in Machine Learning, Neural and Statistical Classification, D. Michie, D. J. Spiegelhalter and C. C. Taylor, Editors 1993,

38. **D. Michie, D. J. Spiegelhalter and C. C. Taylor** (Ed), (1993), "*Machine Learning, Neural and Statistical Classification*",

39. **J. R. Quinlan** (1986) "*Induction of Decision Trees*", Machine Learning, **1**(1), pp. 86-106.

40. **B. Schweizer and A. Sklar** (1961) "*Associative functions and statistical triangle inequalities*", Publ. Math. Debrecen, **8**, pp. 169-186.

41. **J. G. Shanahan** (1996) "*Automatic Synthesis of Fuzzy Rule Cartesian Granule Features from Data for both Classification and Prediction*", (Report No. ITRC 247), Advanced Computing Research Centre, Dept. of Engineering Maths, University of Bristol, UK.

42. **J. G. Shanahan** (1998) "*Cartesian Granule Features: Knowledge Discovery of Additive Models for Classification and Prediction*", PhD Thesis, Dept. of Engineering Maths, University of Bristol, UK,

43. **J. G. Shanahan** (1998) "*Inductive logic programming with Cartesian granule features*", Personal Communication, Dept. of Engineering Maths, University of Bristol, UK.

44. **H. A. Simon** (1983) *"Why should machine learn?"*, in Machine Learning: An Artificial Intelligence Approach, R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Editors 1983, Springer-Verlag. Berlin. pp. 25-37.

45. **J. W. Smith, et al.** (1988), "*Using the ADAP learning algorithm to forecast the onset of diabetes mellitus*", in Proc. Symposium on Computer Applications and Medical Care, , pp 261-265.

46. **M. Sugeno and T. Yasukawa** (1993) "*A Fuzzy Logic Based Approach to Qualitative Modelling*", IEEE Trans on Fuzzy Systems, **1**(1), pp. 7-31.

47. **G. Syswerda** (1989), "*Uniform crossover in genetic algorithms*", in Proc. Third Int'l Conference on Genetic Algorithms, , pp 989-995.

48. **W. A. Tackett** (1995) "*Mining the Genetic Program*", IEEE Expert, (6), pp. 28-28.

49. **R. R. Yager** (1994) "*Generation of Fuzzy Rules by Mountain Clustering*", J. Intelligent and Fuzzy Systems, **2**, pp. 209-219.

50. **L. A. Zadeh** (1994) "*Soft Computing and Fuzzy Logic*", IEEE Software, **11**(6), pp. 48-56.

51. **L. A. Zadeh** (1996) "*Fuzzy Logic = Computing with Words*", IEEE Transactions on Fuzzy Systems, **4**(2), pp. 103-111.