ELSEVIER International Journal of Approximate Reasoning 22 (1999) 109–148

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

Controlling with words using automatically identified fuzzy Cartesian granule feature models

James F. Baldwin, Trevor P. Martin, James G. Shanahan *,1

Department of Engineering Mathematics, University of Bristol, Advanced Computing Research Centre, Queen's Bldg, University Walk, Bristol BS8 1TR, UK

Received 1 October 1998; accepted 1 February 1999

Abstract

We present a new approach to representing and acquiring controllers based upon Cartesian granule features – multidimensional features formed over the cross product of words drawn from the linguistic partitions of the constituent input features – incorporated into additive models. Controllers expressed in terms of Cartesian granule features enable the paradigm "controlling with words" by translating process data into words that are subsequently used to interrogate a rule base, which ultimately results in a control action. 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 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. We illustrate the approach on a variety of problems including the modelling of a dynamical process and a chemical plant controller.

Keywords: Cartesian granule features; Controlling with words; Fuzzy control; Mass assignment theory

* Corresponding author. Tel.: +33-476-61-51-13; fax: +33-476-61-50-99; e-mail: shanahan@xrce.xerox.com

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

PII: S0888-613X(99)00016-X

1. Introduction

Traditionally fuzzy controllers have been acquired directly from experts in the field (i.e. were manually programmed), however this has led to many restrictions.

- The problems that could be modelled using fuzzy controllers were limited to well understood tasks (either mathematically or behaviourally) [2].
- High dimensional problems were generally ignored.
- Problem domains where domain knowledge is overly difficult to capture (too much, too little, too expensive etc...) for example in intelligent activities such as plant control, scheduling etc. could not be addressed effectively.
- Fewer problems could be addressed due to the programming bottleneck (software lag) that results from deploying such systems. Overcoming this issue is seen as one of the most important areas of computer science over the next 20 years.

Automatic acquisition of controllers through machine learning is seen as a means of alleviating many of the problems facing our cyborg society outlined above. Numerous approaches to fuzzy modelling approaches exist. See Refs. [11,16,47,51] for examples, where fuzzy modelling approaches have been applied to control problems (and other problem domains) with high degrees of success. These approaches however, can suffer from various problems including decomposition error, over complex models and local minima models. For example the data browser approach presented in Ref. [10] can suffer from decomposition error [43]. The approaches presented in Refs. [11,16,47,51] provide no natural means of identifying or representing high-dimensional systems. Furthermore, the identification techniques used in these approaches can suffer from local minima models due to the hill-climbing search strategies employed. These issues are not just limited to fuzzy based approaches but also apply to other knowledge representation and identification strategies.

The work presented here tries to overcome these limitations. This is enabled through the use of Cartesian granule features – multi-dimensional features built on words – and a corresponding identification algorithm, G_DACG. 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 [12,13,42,43]. 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, as alluded to earlier, 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 various problems in Section 5 and compare the results obtained with other standard machine learning approaches. Finally we finish 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 type of systems are not considered here however, [43] 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 Refs. [11,15]; 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 Refs. [32,33,49] 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 [26] 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. Refs. [17,30] 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 [18,27] are examples of an embedded feature selection strategy where the model is iteratively refined by modifying, adding or removing features. MARS [22], 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 [17,18,27,30]. 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 Ref. [30] 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 [28] 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 [17]. 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'') [31]. REIGN [16] 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 [26] 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 [30]. 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 Ref. [30] 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 [20,25]. 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 Refs. [47,51] fuzzy partitioning is used. The granularity of the partitions in these approaches is determined heuristically. In the case of [51] 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 that are sub-optimum in nature. In the case of feedforward neural networks [24] 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 [24]. 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 coefficients 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 [47,51]. Once again commonly used procedures such as the mountain method [51] 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 determining the class Cartesian granule fuzzy sets (also selecting suitable granule characterisations) and also 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 [12,13,42,43]. 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* [53,54], 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 $W_1 \times W_2 \times \cdots \times W_m$, where each W_i is a word or label associated with a fuzzy set defined over the universe Ω_i and where "$\times$" 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 $P_1 \times P_2 \times \cdots \times P_m$ where each P_i is a linguistic partition of universe Ω_i and where "×" denotes the Cartesian product. In other words given a set of single attribute features $\{F_1, F_2, \ldots, F_m\}$ defined over $\{\Omega, \Omega, \ldots, \Omega_m\}$, 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.

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

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

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

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, consider the Cartesian granule $w_{11} \times w_{21} \times \cdots \times w_{m1}$, 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 $w_{11} \times w_{21} \times \cdots \times w_{m1}$ is calculated as follows:

$$\mu_{w_{11}}(x_1) \wedge \mu_{w_{21}}(x_2) \cdots \wedge\ \mu_{w_{m1}}(x_m),$$

where x_i is the feature value associated with the ith feature within the data vector. Here the aggregation operator $\wedge$ can be interpreted as any T-norm [29,41] such as *product* or *minimum*. The choice of conjunction operator is considered in Ref. [43].

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

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

This is depicted in Fig. 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.

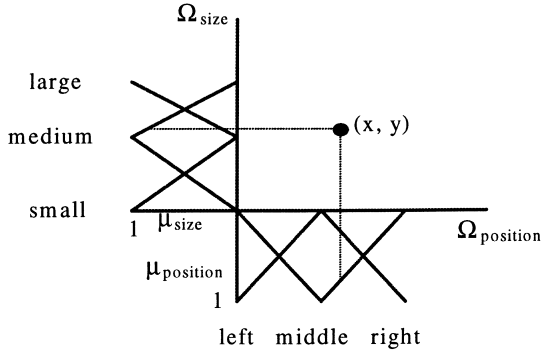

Fig. 1. Fuzzy partitions of universes Ω_{size} and $\Omega_{positiion}$.

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

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 [11] notation) [1] could be

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

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 ∧ as product.

[1] 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*):

$$\mu_{middle}(x) = \begin{cases} 0 & \text{if } x \leqslant 0, \\ x/50 & \text{if } 0 < x \leqslant 50, \\ (100 - x)/50 & \text{if } 50 < x < 100, \\ 0 & \text{if } x \geqslant 100. \end{cases}$$

3.1.2. Mass assignment theory

A brief introduction to mass assignment theory is necessary to appreciate (1) the methods used in the induction of concepts in terms of Cartesian granule fuzzy sets (presented subsequently in Sections 2 and 3.1.3) the inference process used within Cartesian granule feature models (Section 3.1.5). Mass assignment theory has been developed by Baldwin [5,11] to provide a formal framework for manipulating both probabilistic and fuzzy uncertainties.

A mass assignment over a finite frame of discernment Ω is a function

$$m : P(\Omega) \rightarrow [0, 1],$$

where $P(\Omega)$ is the power set of Ω and satisfies the condition

$$\sum_{A \in P(X)} m : (A) = 1.$$

Every set $A \in P(\Omega)$ for which $m(\mathrm{A}) > 0$ is called a *focal element* of m. A mass assignment can be viewed as a form of knowledge that expresses upper and lower probabilities for the individual elements of the frame of discernment. In other words, a mass assignment can be viewed as a family of probability distributions, all of which satisfy the axioms of probability theory and the upper and lower constraints delimited by the mass assignment. Consequently, although mass assignments can represent probabilities they have the added flexibility of being able to represent uncertain probabilities. For example, consider a class of undergraduate students where students can be classified as *first-class honours*, *second-class honours* or as *pass*. Consider the case where there are 100 students, where it is known that 30 are *pass* students, 40 are *second-class honours* or *pass* and the remainder unknown. This can be more succinctly written in mass assignment format as follows.

$$\begin{aligned} \mathrm{MA}_{Class} = {} & \{pass\} : 0.3 \\ & \{pass, second\text{-}class\ honours\} : 0.4 \\ & \{pass, second\text{-}class\ honours, first\text{-}class\text{-}honours\} : 0.3. \end{aligned}$$

This mass assignment corresponds to the following family of probability distributions

$$\begin{aligned} 0.3 &\leqslant \Pr(pass) \leqslant 1 \\ 0 &\leqslant \Pr(second\text{-}class) \leqslant 0.7 \\ 0 &\leqslant \Pr(first\text{-}class) \leqslant 0.3 \end{aligned}$$

such that

$$\Pr(pass) + \Pr(second\text{-}class) + \Pr(first\text{-}class) = 1.0.$$

A particular type of probability distribution is obtained by distributing the mass associated within the non-singleton focal elements uniformly; this

distribution is termed as the *least prejudiced distribution* (LPD) [5]. In the case of MA_{Class} the corresponding LPD, LPD_{Class} is given as follows.

$$\Pr(pass) = 0.3 + 0.4/2 + 0.3/3 = 0.6$$
$$\Pr(second\text{-}class) = 0.4/2 + 0.3/3 = 0.3$$
$$\Pr(first\text{-}class) = 0.3/3 = 0.1.$$

The transformation of mass assignment to a least prejudiced distribution is reversible; hence given a least prejudiced distribution it is possible to find a corresponding mass assignment.

Mass assignments are related to fuzzy sets via the voting model [3]. Consider that a variable V has a fuzzy set value f as follows:

$$V \text{ is } f,$$

where f is a fuzzy set defined on the discrete universe $X = \{x_1, x_2, \ldots, x_n\}$ written more succinctly as follows:

$$f = \sum_{i=1}^{n} x_i/\chi_i.$$

This proposition that "V has a fuzzy set value f" induces a possibility distribution over the values of X such that the membership values of x_i are numerically equated with possibility i.e.

$$\Pi(x_i) = \chi_i.$$

Suppose f is a normalised fuzzy set whose elements are ordered such that

$$\chi_1 = 1, \quad \chi_i \leqslant \chi_j \quad \text{if } i < j,$$

then

$$\Pi(\{x_i, \ldots, x_n\}) = \chi_i$$

so with the assumption that $\Pr(A) \leqslant \Pi(A)$ for any $A \in P(X)$, (where $P(X)$ corresponds to the power set of X) we can find the mass assignment corresponding to the fuzzy set f is as follows

$$MA_f = \{\{x_1, \ldots, x_i\} : \chi_i - \chi_{i+1}\} \quad \text{with } \chi_{n+1} = 0.$$

This can be extended to non-normal fuzzy sets so that the mass assignment corresponding to the fuzzy set f looks like the following

$$MA_f = \{\{x_1, \ldots, x_i\} : \chi_i - \chi_{i+1}, \{\emptyset\} 1 - \chi_i\} \text{ with } \chi_{n+1} = 0,$$

such that a non-zero mass is assigned to the null set, in this case the mass assignment is said to be incomplete. The extension to fuzzy sets over continuous universes is a little more involved and is achieved by taking alpha cuts of the fuzzy set and proceeding in a similar fashion as described above with continuous integrals.

The relationship between probability and possibilities has been investigated by others including [21,46,52]. Since the focal elements in the mass assignment corresponding to a fuzzy set are nested (consonant) there exists a straightforward transformation from fuzzy sets to mass assignments to frequency distributions. This bi-directional transformation plays a vital role in the learning algorithms presented in subsequent sections, facilitating learning through a counting approach over granules/words and subsequent knowledge expression in a transparent/intuitive fuzzy set format. Here we have merely presented the discrete case; the continuous case is similar and is illustrated in Ref. [11].

3.1.2.1. Semantic unification. In Refs. [4,5] a detailed presentation of the mass assignment calculus (meet, join, restrictions, conditioning) is presented. In this paper we are concerned mainly with the conditioning operation – semantic unification. Semantic unification is a conditioning operation between two mass assignments [7]. This operation provides a very natural and formal means of measuring the degree of "match" between concepts expressed in terms of fuzzy sets, made possible through the bi-directional mapping between fuzzy sets and mass assignments. The maintenance of the uncertain nature of the probabilities in a mass assignment following conditioning, corresponds to two versions of semantic unification: interval and point-valued. Interval semantic unification maintains the uncertainty of probabilities present in the original mass assignments through its interval representations, corresponding to an upper and lower bounds of the conditional probability, whereas point semantic unification corresponds to the expected value of the membership of a fuzzy set f given the least prejudiced distribution (LPD) of fuzzy set g [8]. This is expressed more succinctly as follows for the discrete case

$$\mathrm{SU}(f,g) = \Pr(f \mid g) = \sum_{i=1}^{n} \chi_f(x_i) \times \mathrm{LPD}_g(x_i),$$

where both the fuzzy set f and g are defined over the universe $X = \{x_1, x_2, \ldots, x_n\}$.

The continuous case is as follows

$$\mathrm{SU}(f,g) = \Pr(f \mid g) = \int_{x \in X} \chi_f(x) \times \mathrm{LPD}_g(x)\, \mathrm{d}x,$$

where f and g are both defined over the continuous universe X. In this paper the use of semantic unification is restricted to point-valued unification, although future work could harness the more expressive interval unification.

3.1.3. 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:

$$\langle \vec{i},\ \text{output} \rangle$$

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 $\Omega_1, \Omega_2, \ldots, \Omega_m$. Here we have extended the notion of a conventional database attribute value to the case where uncertain or vague information can be 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) [43] and build Cartesian granule fuzzy sets corresponding to each concept (fuzzy set) in the output space.

3.1.3.1. 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 two 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.

3.1.3.2. 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} corresponding to this fuzzy set CGF_{Ci} via its mass assignment (see Section 3.1.2). 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 [11]. The induction algorithm for prediction problems is described in detail in Ref. [13] and is summarised in Fig. 2.

3.1.4. Additive Cartesian granule feature models

Refs. [14,43] 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 [6,11] provide a natural mechanism for capturing this type of decomposed approach to systems modelling [43] 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 captures very naturally additive Cartesian granule feature models. A simplified evidential logic rule structure is depicted in Fig. 3. Here *CLASS* can be viewed as a fuzzy set consisting of a single crisp value, in the case of classification type problems, or as a fuzzy set characterising part of the output variable universe in the case of prediction problems. Each rule characterises the relationship between input and output data for a particular region of the output space i.e. a concept. A rule is generated for each class in the output space.

INDUCTION ALGORITHM FOR PREDICTION PROBLEMS

Select base input features which will be combined to form a Cartesian granule feature
Determine granularity and granule characterisation for each base feature
Create linguistic partitions over each of the base feature universes
Form the Cartesian Granule Universe Ω_{CG}
Form a fuzzy partition of the output variable universe

Repeat for each fuzzy class $\tilde{C}$ in the output space
  Initialise a frequency distribution $DIST_{CG}$ over the Cartesian Granule Universe Ω_{CG}
  Repeat for each data tuple T_i in the training database whose output value has a membership value > 0 (i.e. $\mu_{\tilde{C}}(outputValueT_i) > 0$)
    Construct the corresponding Cartesian granule fuzzy set CGF_{Ci}
    Form the Least Prejudiced Distribution LPD_{Ci} corresponding to this fuzzy set CGF_{Ci} via MAT. This corresponds to a frequency distribution.
    Update overall frequency distribution $DIST_{CG}$ with LPD_{Ci} in proportion to $\mu_{\tilde{C}}(outputValueT_i)$.
  End Repeat
  Using MAT construct the unique fuzzy set corresponding $DIST_{CG}$
End Repeat

Fig. 2. Outline of Cartesian granule fuzzy set induction algorithm for prediction problems.

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

Fig. 3. Evidential logic rule structure.

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.

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.

3.1.5. Inference and decision making

Here we consider the general *inference* and *decision-making* processes used within this framework of knowledge representation. 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 3.1.2.1) which results in the Cartesian granule fuzzy set description CGD of $\vec{x}$. Then we merely carry out the semantic unification (SU) between the class fuzzy set CGF and the data fuzzy set CGD. In other-words, in the case of Cartesian granule features

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

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

$$B_{Class} = \sum_{i=1}^{m} \mathrm{SU}(f_{iClass} \mid \vec{x}_i) w_{iClass},$$

where W_{iClass} is the weight of importance associated with feature i for class *Class*.

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. In the case of prediction problems, the process of predicting of the value of the output variable associated with the input data vector is known as defuzzification. There are many standard defuzzification procedures such as Centre of Area (COA), Centre of Gravity (COG) [29] which could be utilised here. However we choose a procedure which incorporates the spirit of the mass assignment theory. The results of inference is a collection of rule hypotheses of the form (*outputFeature*$_i$ *is fuzzyNumber*$_i$): (α_i) which have non-zero supports α_i. In this case *fuzzyNumber*$_i$ is a fuzzy set defined over the universe of the output variable. Our defuzzification procedure involves firstly calculating the expected value of least prejudiced distribution associated with each fuzzy set f_i via the mass assignment associated with f_i [11]. This yields a collection of values v_i and the supports from their respective head clauses as follows (v_i):(α_i). We then take the expected value of these values, which yields a specific point value, i.e. the result of our inference. In other words we calculate the inferred point value as follows

$$v = \sum_{i=1}^{m} v_i \alpha_i,$$

where v_i is the expected value of the LPD associated with fuzzy set f_i. The value v_i corresponds to the predicted output value for the system. In the case where the sum of all the rule supports is less than 1 (i.e. $\sum_{i=1}^{m} \alpha_i < 1$), the rule supports, α_i, are normalised as follows.

$$\frac{\alpha_i}{\sum_{j=1}^{m} \alpha_j}.$$

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 [32,33] 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 [13,42]. 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 [43]. However empirical evidence on various problem domains seems to suggest that there is very little difference between the effectiveness of both these operators [13,42]. Consequently we have reduced our function set to the product operator CG*Product*. At a later date it is hoped to allow a richer function set and genetically select appropriate conjunction operators. The arity of the CG*Product* 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 system modelling along with their respective granularity range (abstraction). For example if we have two 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, CG*Product*, 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

$$\mathrm{Discrimination}_i = 1 - \max_{\substack{j=1 \\ j \neq i}}^{C} \mathrm{SU}(\mathrm{CGF}_i \mid \mathrm{CGF}_j),$$

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 Fig. 4.

We combine the individual factors in the following manner:

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

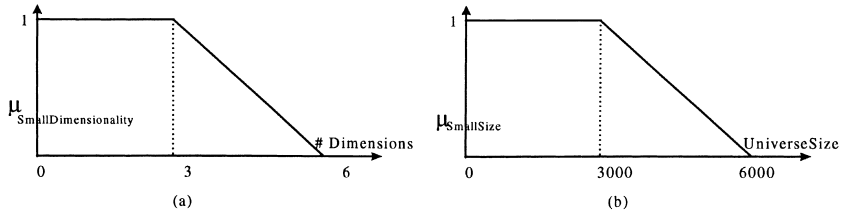

Fig. 4. (a) Fuzzy set corresponding to small dimensionality in Cartesian granule features. (b) Fuzzy set corresponding to small size of Cartesian granule feature universes.

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 Ref. [23]. 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:

$$\sum_{\text{granularity}=\text{minGran}}^{\text{max\_Gran}} \sum_{\text{dim}=1}^{\text{MaxDim}} {}^{\text{Num Of Features}}C_{\text{dim}} * (\text{granularity})^{\text{dim}}.$$

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) [32,48]. 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 [48]. 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 Fig. 5 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.

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

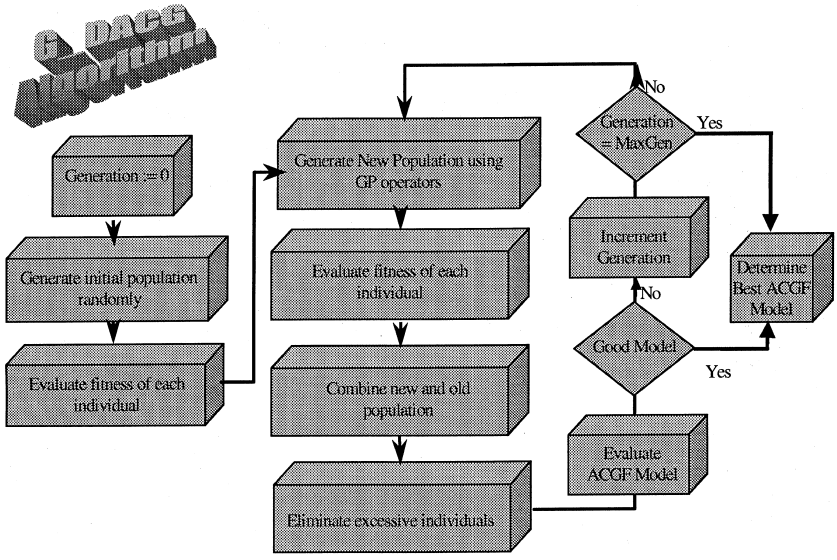

Fig. 5. G_DACG constructive induction algorithm.

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 [20], 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 Ref. [43] 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 Ref. [43].

5. Results

The G_DACG algorithm has been illustrated and compared with other machine learning approaches on a variety of problem domains including object recognition [43]. Here we illustrate the approach on some benchmark machine learning problems and control problems.

5.1. Ellipse problem

The ellipse problem serves as a simple illustration of the G_DACG algorithm from a classification problem perspective. 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 Fig. 6 for the ellipse inequality

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

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 $\langle X, Y, Class\rangle$, where *Class* adopts the value 0 for *illegal* indicating that the point $\langle X, Y\rangle$ 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

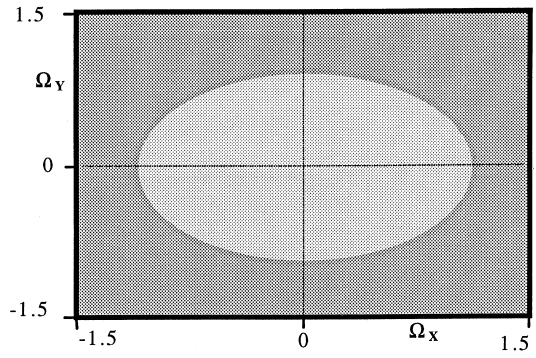

Fig. 6. Ellipse inequality in Cartesian space. Points in lightly shaded region satisfy the ellipse inequality and thus are classified as legal. Points in darker region are classified as illegal.

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 [32] 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 three 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 Fig. 7. The rule corresponding the legal class consists of three Cartesian granule features, while the rule for the illegal case consists of just two features. Backward elimination based upon semantic discrimination eliminated the third feature from the illegal rule. The optimally determined filters correspond to the "true" filter (i.e. the identity filter

```
((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  ) ) )

((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 ) ) )
```

Fig. 7. An example of an additive Cartesian feature model in Fril for the ellipse problem. This model gives over 98.7% accuracy on test cases.

$f(x) = x$)) for this model (not shown in Fig. 7). 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.

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 with 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.

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 [9]	$X,\ Y$	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.

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 four 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

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 h serum insulin (mu/U/ml)
5	Body mass index (kg/m^2)
6	Diabetes pedigree function
7	Age (years)

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 Fig. 8. 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.

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

```
?((def_itype POSITIVE_FILTER [0:0.0 1:1 ]))
?((def_itype NEGATIVE_FILTER [0:0 0.79:1 ]))

((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  ) ) )
```

Fig. 8. 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.

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.

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).

5.3. $\sin(X * Y)$ *prediction problem*

In the previous sections we have illustrated the effectiveness of Cartesian granule features in modelling classification systems i.e. systems where the dependent output variable is discrete in nature. Here however, we address prediction problems (also known as function approximation) i.e. whose dependent output variable is continuous in nature. We demonstrate the effectiveness with which Cartesian granule features can model a non-linear static system; based on the function $\sin(X * Y)$. The $\sin(X * Y)$ function (swan's neck) has two base input variables, X and Y and is graphically depicted in Fig. 9. The considered domain for both the X and Y variable, is [0, 3]. Different training, control (validation) and test datasets, comprising of 529 (in grid fashion), 600 (randomly) and 900 (in grid fashion) data vectors, respectively, were generated. Each data sample consists of a triple $\langle X, Y, \sin(X * Y)\rangle$.

Table 3
Comparison of results for the Pima diabetes detection problem

Approach	Accuracy (%)
Additive Cartesian granule feature model	79.7
Mass assignment based MATI [9]	79.7
Oblique decision trees [43]	78.5
Neural net (normalised Data)	78
C4.5 [38]	73
Data browser	70

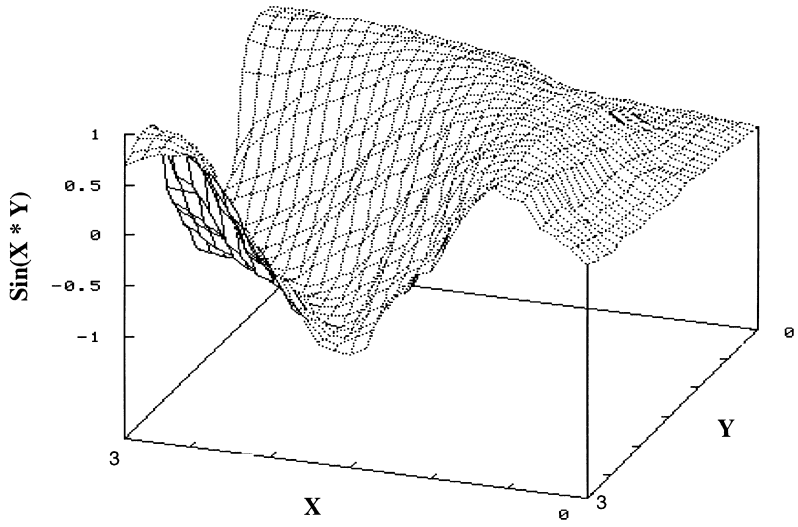

Fig. 9. Graphic representation of $\sin(X * Y)$.

5.3.1. Additive Cartesian granule feature modelling of $\sin(X * Y)$ *problem*

We applied the G_DACG constructive induction algorithm to the $\sin(X * Y)$ problem. Both base features were considered and Cartesian granule features of dimensionality up to five with granularity ranges of [2,14] were considered (while parsimony was promoted in the form of the fitness function used). The k-tournament selection parameter k was set to three for this problem. The G_DACG algorithm iterated for thirty generations (or if the stopping criterion was satisfied it halted earlier, arbitrarily set at an RMS error (root mean square error) of 1%). The output universe was partitioned using both triangular and trapezoidal fuzzy sets, both uniformly placed or positioned as a result of percentile partitioning (i.e. placing roughly equal numbers of examples in each fuzzy set), using a variety of granularities. Additive Cartesian granule feature models consisting of three Cartesian granule features were constructed at various stages during the G_DACG process. Backward elimination based on fitness was also employed. As a result of the G_DACG process an additive Cartesian granule feature model where each rule consists of a single two dimensional Cartesian granule feature was deemed to be the most suitable model. The granularities of the base feature universes in this case were 14 and 14. The input granule characterisation was a trapezoidal fuzzy set with an overlap rate of 40%. On the other hand the granularity of the output universe was set to 6 where each granule was characterised by a triangular fuzzy set. The output fuzzy sets were uniformly distributed over the variable universe. The discovered model yielded an RMS error of 4.12%. The decision surface for this model plotted against the actual surface is depicted in Fig. 10. When the

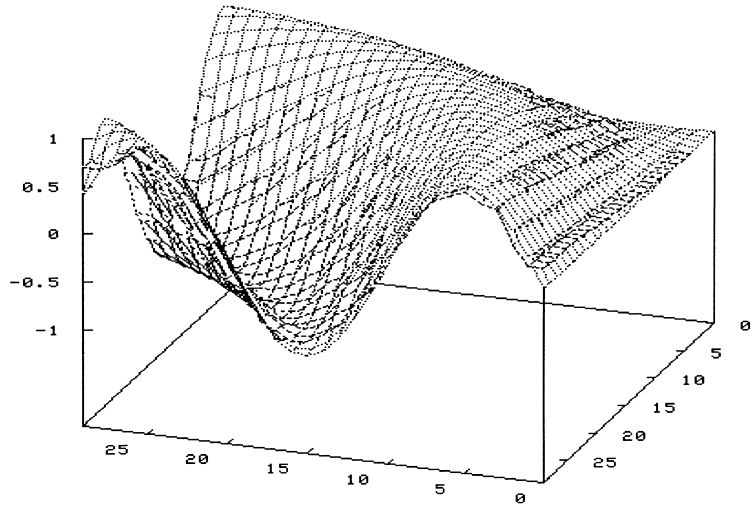

Fig. 10. $\sin(X * Y)$ decision surface generated using two-dimensional Cartesian granule features, where the base feature universes have been partitioned using 14 trapezoidal fuzzy sets with 40% overlap. The RMS error is 4.12%.

granularity of the input base features is increased to 20 an RMS error of 2.6% is achieved.

5.3.2. $\sin(X * Y)$ *problem results comparison*

The use of various types of Cartesian granule features incorporated into rules has been investigated in the context of the $\sin(X * Y)$ prediction problem. Both one-dimensional and multidimensional Cartesian granule features were investigated on an individual basis and when incorporated into additive models. The use of strictly one-dimensional feature models for this problem yields a high decomposition error. Cartesian granule features where the underlying granules are characterised by crisp sets can not be used to model prediction problems and generally the crisper fuzzy sets tended to perform badly in modelling this problem [43]. Modelling approaches which use decomposition including the one-dimensional Cartesian granule feature models and the data browser suffer from large decomposition errors. A results summary of each of the examined approaches is presented in Table 4. Overall additive Cartesian granule feature models lead to high levels of accuracy for this problem which also providing good model transparency.

5.4. Modelling a dynamical system – the Box–Jenkins gas furnace problem

This example deals with the widely used benchmark problem of modelling a gas furnace (an example of a dynamical process) which was first presented by

Table 4
Summary of $\sin(X * Y)$ prediction problem results using various supervised learning approaches

Approach	Features details	RMS error
Neural network	X, Y, and 7 hidden nodes	2.19
Two-dimensional Cartesian granule feature	(X, Y), granularity = 20, triangular fuzzy set, output granularity = 6	2.6
Two-dimensional Cartesian granule feature	(X, Y), granularity = 14, 40% overlapping trapezoids, output granularity = 6	4.12
MATI Decision Trees [9]	X, Y Trapezoidal fuzzy sets with overlap degree of 50%, Output granularity = 5.	4.2
One-dimensional Cartesian granule features in evidential rules	(X), (Y), regardless of granularity of input or output spaces	25+
Data browser (conjunctive logic rules)	X, Y (non-smoothed fuzzy sets) regardless of granularity of output universe.	23+

Box and Jenkins [19]. The modelled system consists of a gas furnace in which air and methane are combined to form a mixture of gases containing CO_2 (carbon dioxide). Air fed to the furnace is kept constant, while the methane feed rate can be varied in any desired manner. The furnace output, the CO_2 concentration, is measured in the exhaust gases at the outlet of the furnace.

The dataset here corresponds to a time series consisting of 296 successive pairs of observations of the form $(u(t), y(t))$, where $u(t)$ represents the methane gas feed rate at the time step t and $y(t)$ represents the concentration of CO_2 in the gas outlets. The sampling time interval is nine seconds. Using a time-discrete formulation, the dynamics of the system is represented by a relationship that links the predicted system state $y(t+1)$ to the previous input states $u(t_i)$ and the previous output states $y(t_i)$, that is $y(t+1)$ is a function of the previous input and output states i.e. $y(t+1) = f(u(t_1), u(t_2), \ldots, u(t_n), y(t_1), y(t_2), \ldots, y(t_n))$. Here we have set the value of n to five. Consequently we consider ten input variables and our database reduces to 291 data tuples of the form $(u(t), u(t-1), \ldots, u(t-4), y(t), y(t-1), \ldots, y(t-4+1))$.

5.4.1. Additive Cartesian granule feature modelling of the gas furnace problem

In the case of this problem all data tuples were considered for both training and testing. The main reason for this is provide a comparison with other approaches presented in the literature. We applied the G_DACG constructive induction algorithm to the gas furnace problem. All ten 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 four for this problem. The output universe was uniformly partitioned using eight triangular fuzzy sets. The G_DACG algorithm iterated for fifty generations (or if the stopping criterion was satisfied it halted earlier, arbitrarily set at an mean square error (MSE) of less than 0.05). As a result of the G_DACG process an additive Cartesian granule feature model where each rule consists of two Cartesian granule features was deemed to be the most suitable model. The model consists of eight rules and a trapezoidal fuzzy set with 50% overlap was determined to be the best input feature granule characterisation. The performance accuracy of the model was measured based upon the mean square error (MSE) between the actual data outputs and the model outputs, that is,

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y^i - \hat{y}^i)^2.$$

The discovered model yields a relatively low MSE of 0.128. In Fig. 11 the model performance is compared with the original data. The Fril code corresponding to a rule in this model is presented in Fig. 12. Increasing the granularity of the output universe (and consequently the number of rules) can lead to models with lower MSE, however, this also leads to more complex models. For example, if the granularity of the output universe is increased to ten the MSE of the model drops to 0.11.

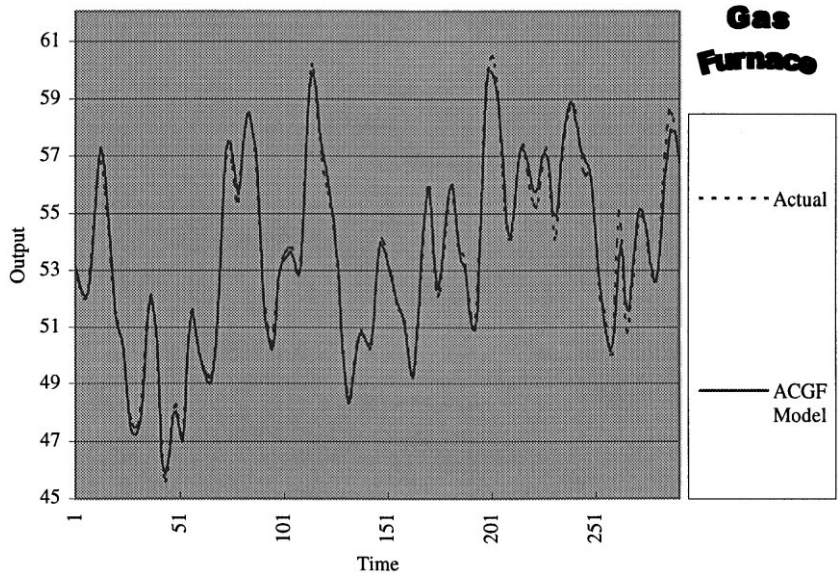

Fig. 11. ACGF model predictions versus the actual data for gas furnace problem.

```
((Predicted level of CO2 at time(t+1) is small)
 (evlog identityFilter (
  (cgValue of ((x(t-3) 10) (y(t) 10))
              in case (CASE) correspond to smallClass) .49
  (cgValue of ((x(t-2) 10) (y(t) 10))
              in case (CASE) correspond to smallClass) .51 ) ) )
```

Fig. 12. An example of a rule in the ACGF model for the gas furnace problem. This model yields an MSE of 0.128. Here identityFilter corresponds to $f(s) = s$.

5.4.2. Gas furnace results comparison

The gas furnace problem serves as a benchmark problem in the field of system identification and has been tested on many learning approaches. Table 5 compares some of the results of the more common statistical and fuzzy based techniques with the ACGF modelling approach. Overall the ACGF modelling approach generally outperforms the other fuzzy and statistical based approaches from an accuracy perspective. The Takagi-Sugeno linear model gives the best performance accuracy, however it lacks the transparency provided by the other approaches including that of ACGF modelling. The models generated by the various approaches were evaluated on the same data that was used to generate them. As a result the results provided no information on the generalisation powers of the extracted models. From a model transparency, the extracted ACGF model is relatively easy to interpret since the extracted Cartesian granule fuzzy sets are all two dimensional in nature, where the Cartesian granules are used to describe/characterise the various concept fuzzy sets. The various fuzzy approaches listed in Table 5, differ mainly in the identification algorithms used. In general, they use local hill climbing strategies and treat the steps of input variable selection and abstraction separately, which may subsequently result in models which are only locally optimum.

Table 5
Comparison of results for the gas furnace problem

Approach	MSE
Box & Jenkins statistical (1970) approach [19]	0.710
Tong (1980) fuzzy model [50]	0.469
Pedrycz (1984) fuzzy model [40]	0.320
Linear model [47]	0.193
Takagi–Sugeno linear model (1993) [47]	0.068
Fuzzy position gradient model (1993) [47]	0.190
Nakoula et al's. fuzzy model (1997) [39]	0.175
Additive Cartesian granule feature model	0.128

5.5. Modelling human operation of a chemical plant controller

Here we generate a model of an operator's control of a chemical plant. This problem and corresponding dataset is presented in Ref. [47]. The chemical plant produces a polymer by the polymerisation of some monomers. Since the start-up of the plant is very complicated, a human operator is required to manually control the plant.

The dataset consists of 70 observations taken from actual plant operation. Each observation consists of five input variables (see Table 6 for details) and an output variable corresponding to the set point for monomer flow rate. The human operator determines the set point for the monomer flow rate and gives this information to a PID controller, which calculates the actual monomer flow rate for the plant.

5.5.1. Additive Cartesian granule feature modelling of the chemical plant problem

In the case of this problem all data tuples were considered for both training and testing. The G_DACG constructive induction algorithm was applied to the chemical plant problem, where all the base input features were considered and Cartesian granule features of dimensionality up to five with granularity ranges of [2,12] were considered. The k-tournament selection parameter k was set to four for this problem. The output universe was uniformly partitioned using eight triangular fuzzy sets. The G_DACG algorithm iterated for fifty generations (or if the stopping criterion was satisfied it halted earlier, arbitrarily set at a root mean square error (RMS) of less than 0.05%). As a result of the G_DACG process an additive Cartesian granule feature model where each rule consists of a single Cartesian granule features was deemed to be the most suitable process controller. The model consists of eight rules and a trapezoidal fuzzy set with 5% overlap was determined to be the best input feature granule characterisation. The performance accuracy of the model was measured based upon the root mean square error (RMS). The discovered model yields an RMS of 2%. The Fril code corresponding to a rule in this model is presented in

Table 6
Input and output base features for chemical plant control problem

No.	Class
0	Monomer concentration
1	Change of monomer concentration
2	Monomer flow rate
3, 4	Local temperatures inside plant
5	Set point for monomer flow rate

```
((Predicted level of y is small)
 (evlog identityFilter (
  (cgValue of ((f0 14) (f1 14) (f2 13) (f3 5))
              in case (CASE) correspond to smallClass) 1 ) )
```

Fig. 13. An example of a rule in the ACGF model for the chemical plant problem. This model yields an RMS of 2%.

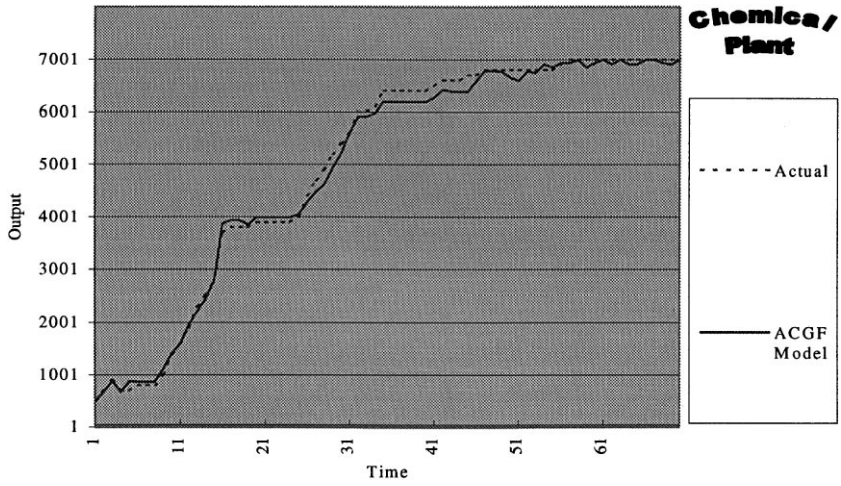

Fig. 14. ACGF model predictions versus human operator for the chemical plant.

Fig. 13. In Fig. 14 the model performance is compared with that of the human operator.

5.5.2. Chemical plant results comparison

Overall the generated additive Cartesian granule feature model performs very well when compared to the human operator. In the case of this problem eight rules have been generated to qualitatively describe the behaviour of the plant. When a neural network is generated using the same input features as the ACGF model and with 4 hidden nodes an RMS error of 1% is achieved.

The discovered ACGF model has a high complexity in this case and may be suffering from the uniform partitioning of the input feature universes. A more efficient and possibly a lower dimensional Cartesian granule feature may result from a data centred approach to partitioning. This is indicated by the number of activated Cartesian granules in the individual fuzzy sets which is for most concepts around ten. Currently other partitioning approaches such as clustering are been investigated.

6. Conclusions

We have presented a new approach to representing and acquiring automatically, controllers from example data based upon Cartesian granule features incorporated into additive models. Controllers expressed in terms of Cartesian granule features, enable the paradigm "controlling with words" by translating process data into words that are subsequently used to interrogate a rule base, and ultimately result in a control action. A corresponding constructive induction algorithm – G_DACG – which automatically identifies additive Cartesian granule feature models was also presented. 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 and gas furnace problems, needs further work when applied to real world problems. This is highlighted by the models discovered in the chemical plant problem where the Cartesian granule feature is of high dimensionality and consists of relatively high granularity. Cartesian granule features do however lay the foundations for a learning paradigm that provides high accuracy, while also achieving model transparency. Current work [44] is addressing the transparency issue as follows:

- Increase the expressiveness of the hypothesis language from attribute-value to relational – leading to relational controllers.
- 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.
- More natural and efficient means of representing attribute universes can be achieved using data centred approaches such as clustering and quadtrees.

References

[1] H. Almuallim, T.G. Dietterich, Learning with irrelevant features, in: Proceedings fo AAAI-91, Anaheim, CA, 1991, pp. 547–552.

[2] R. Babuska, Fuzzy modelling: principles, methods and applications, in: C. Bonivento, C. Fantuzzi, R. Rovatti (Eds.), Fuzzy Logic Control: Advances in Methodology, World Scientific, Singapore, 1998, pp. 187–220.

[3] J.F. Baldwin, Combining evidences for evidential reasoning, Int. J. Intelligent Systems 6 (6) (1991) 569–616.

[4] J. F. Baldwin, A Theory of Mass Assignments for Artificial Intelligence, in: A.L. Ralescu (Ed.), IJCAI '91 Workshops on Fuzzy Logic and Fuzzy Control, Sydney, Australia, Lecture Notes in Artificial Intelligence, 1991, pp. 22–34.
[5] J.F. Baldwin, Fuzzy and Probabilistic Uncertainties, in: Shapiro (Ed.), Encyclopaedia of AI, 2nd ed., 1992, pp. 528–537.
[6] J.F. Baldwin, Evidential support logic, FRIL and cased base reasoning, Int. J. Intelligent Systems 8 (9) (1993) 939–961.
[7] J.F. Baldwin, Probabilistic, Fuzzy and Evidential Reasoning in FRIL (fuzzy relational inference language), in: Proceedings of the Two Decades of Fuzzy Control, IEE, London, 1993, pp. 7/1–7/4.
[8] J.F. Baldwin, J. Lawry, T.P. Martin, Efficient Algorithms for Semantic Unification, in: Proceedings of the IPMU, Granada, Spain, 1996, pp. 527–532.
[9] J.F. Baldwin, J. Lawry, T.P. Martin, Mass assignment fuzzy ID3 with applications, in: Proceedings of the Fuzzy Logic: Applications and Future Directions Workshop, London, UK, 1997, pp. 278–294.
[10] J.F. Baldwin, T.P. Martin, Fuzzy Modelling in an Intelligent Data Browser, in: Proceedings of the FUZZ-IEEE, Yokohama, Japan, 1995, pp. 1171–1176.
[11] J.F. Baldwin, T.P. Martin, B.W. Pilsworth, FRIL – Fuzzy and Evidential Reasoning in A.I., Research Studies Press, Wiley, New York, ISBN 086380159 5, 1995.
[12] J.F. Baldwin, T.P. Martin, J.G. Shanahan, Modelling with Words using Cartesian Granule Features. Report No. ITRC 246, Advanced Computing Research Centre, Dept. of Engineering Maths, University of Bristol, UK, 1996.
[13] J.F. Baldwin, T.P. Martin, J.G. Shanahan, Modelling with words using Cartesian granule features, in Proceedings of the FUZZ-IEEE, Barcelona, Spain, 1997, pp. 1295–1300.
[14] J.F. Baldwin, T.P. Martin, J.G. Shanahan, Aggregation in Cartesian granule feature models, in: Proceedings of the IPMU, Paris, 1998, p. 6.
[15] J.F. Baldwin, B.W. Pilsworth, Genetic Programming for Knowledge Extraction of Fuzzy Rules, in: Proceedings of the Fuzzy Logic: Applications and Future Directions Workshop, London, UK, 1997, pp. 238–251.
[16] A. Bastian, Modelling and Identifying Fuzzy Systems under varying User Knowledge, Ph.D thesis, Meiji University, Tokyo, 1995.
[17] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.
[18] K.M. Bossley, Neurofuzzy modelling approaches in system identification, Ph.D thesis, Department of Electrical and Computer Science, Southampton University, Southampton, UK, 1997.
[19] G.E. Box, G.M. Jenkins, Time series analysis forecasting and control, Holden Day, San Francisco, CA, 1970.
[20] P.A. Devijer, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, 1982.
[21] D. Dubois, H. Prade, Fuzzy sets in approximate reasoning one-inference with possibility distributions, Fuzzy Sets and Systems 40 (1991) 143–202.
[22] J.H. Friedman, Multivariate adaptive regression splines, The Annals of Statistics 19 (1991) 1–141.
[23] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural computation 4 (1992) 1–58.
[24] J. Hertz, K. Anders, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, New York, 1994.
[25] A.G. Ivanhnenko, Polynomial theory of complex systems, IEEE Trans. Systems, Man and Cybernetics 1 (4) (1971) 363–378.
[26] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.

[27] T. Kalvi, ASMOD: an algorithm for adaptive spline modelling of observation data, Int. J. Control 58 (4) (1993) 947–968.
[28] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of nineth Conference in Machine Learning, Aberdeen, Scotland, 1992, pp. 249–256.
[29] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Theory and Applications, Prentice-Hall, New Jersey, 1995.
[30] R Kohavi, G.H. John, Wrappers for feature selection, Artificial Intelligence 97 (1997) 273–324.
[31] I. Kononenko, S.J Hong, Attribute selection for modelling, FGCS Special Issue in Data Mining, March/April 1998, pp. 34–55.
[32] J.R. Koza, Genetic Programming, MIT Press, Massachusetts, 1992.
[33] J.R. Koza, Genetic Programming II, MIT Press, Massachusetts, 1994.
[34] L. Ljung, System identification: theory for the user, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[35] C.J. Merz, P.M. Murphy, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html.] University of California, Irvine, CA, 1996.
[36] R.S. Michalski et al., Learning patterns in images, in: R.S. Michalski, I. Bratko, M. Kubat (Eds.), Machine Learning and Data Mining, Wiley, New York, 1998, pp. 241–268.
[37] D. Michie, D.J. Spiegelhalter, C.C. Taylor, Dataset descriptions and results, in: D. Michie, D.J. Spiegelhalter and C.C. Taylor (Eds.), Machine Learning Neural and Statistical Classification, 1993, pp. 131–174.
[38] D. Michie, D.J. Spiegelhalter, C.C. Taylor, Machine Learning, Neural and Statistical Classification, Ellis Horwood, New York, USA, 1993.
[39] Y. Nakoula, S. Galichet, L. Foulloy, Identification of linguistic fuzzy models based on learning, in: H. Helledoorn, D. Driankov (Eds.), Fuzzy Model Identification, Springer, Berlin, 1997, pp. 281–319.
[40] W. Pedrycz, An identification algorithm in fuzzy relational systems, Fuzzy Sets and Systems 13 (1984) 153–167.
[41] B. Schweizer, A. Sklar, Associative functions and statistical triangle inequalities, Publ. Math. Debrecen 8 (1961) 169–186.
[42] J.G. Shanahan, 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, 1996.
[43] J.G. Shanahan, Cartesian Granule Features: Knowledge Discovery of Additive Models for Classification and Prediction, Ph.D thesis, Dept. of Engineering Maths, University of Bristol, Bristol, UK, 1998.
[44] J.G. Shanahan, Inductive logic programming with Cartesian granule features, 1998, in preparation.
[45] J.W. Smith et al., Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Symposium on Computer Applications and Medical Care, 1988, pp. 261–265.
[46] T. Sudkamp, On probability-possibility transformation, Fuzzy Sets and Systems 51 (1992) 73–81.
[47] M. Sugeno, T. Yasukawa, A fuzzy logic based approach to qualitative modelling, IEEE Trans. Fuzzy Systems 1 (1) (1993) 7–31.
[48] G. Syswerda, Uniform crossover in genetic algorithms, J.D. Schaffer, Third Int'l Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, USA, 1989, pp. 989–995.
[49] W.A. Tackett, Mining the genetic program, IEEE Expert 6 (1995) 28.
[50] R.M. Tong, The evaluation of fuzzy models derived from experimental data, Fuzzy Sets and Systems 4 (1980) 1–12.
[51] R.R. Yager, Generation of fuzzy rules by mountain clustering, J. Intelligent and Fuzzy Systems 2 (1994) 209–219.

[52] L.A. Zadeh, Probability measures of fuzzy events, J. Math. Anal. Appl. 23 (1968) 421–427.
[53] L.A. Zadeh, Soft computing and fuzzy logic, IEEE Software 11 (6) (1994) 48–56.
[54] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Trans. Fuzzy Systems 4 (2) (1996) 103–111.