Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.3, 583–601

ROUGH SET-BASED DIMENSIONALITY REDUCTION FOR SUPERVISED AND UNSUPERVISED LEARNING

Qiang SHEN*, Alexios CHOUCHOULAS*

The curse of dimensionality is a damning factor for numerous potentially powerful machine learning techniques. Widely approved and otherwise elegant methodologies used for a number of different tasks ranging from classification to function approximation exhibit relatively high computational complexity with respect to dimensionality. This limits severely the applicability of such techniques to real world problems. Rough set theory is a formal methodology that can be employed to reduce the dimensionality of datasets as a preprocessing step to training a learning system on the data. This paper investigates the utility of the Rough Set Attribute Reduction (RSAR) technique to both supervised and unsupervised learning in an effort to probe RSAR's generality. FuREAP, a Fuzzy-Rough Estimator of Algae Populations, which is an existing integration of RSAR and a fuzzy Rule Induction Algorithm (RIA), is used as an example of a supervised learning system with dimensionality reduction capabilities. A similar framework integrating the Multivariate Adaptive Regression Splines (MARS) approach and RSAR is taken to represent unsupervised learning systems. The paper describes the three techniques in question, discusses how RSAR can be employed with a supervised or an unsupervised system, and uses experimental results to draw conclusions on the relative success of the two integration efforts.

Keywords: knowledge-based systems, fuzzy rule induction, rough dimensionality reduction, knowledge acquisition

1. Introduction

With the widening availability and small size of modern computer systems, intelligent learning systems are rapidly gaining popularity for wide ranges of applications. Learning systems have found their way to all manners of application domains: the stock market, financial customer modelling and risk assessment, industrial monitoring and control, assembly robotics, global and personal information retrieval and filtering, and even computer games. This success is easily explained by the fact that learning systems are cost-effective when they are applicable. The price of computing equipment has dropped dramatically over the past decade, while the time of human experts has remained steadily expensive. Having even a fraction of the knowledge of a highly paid and competent consultant built into a computation system is clearly very desirable.

* Institute for Representation and Reasoning, Division of Informatics, The University of Edinburgh, Edinburgh EH1 1HN, U.K. e-mail: {qiangs,alexios}@dai.ed.ac.uk

In addition, a system that learns automatically from historical data typically works faster than a human expert. For instance, in cases like information retrieval and filtering (van Rijsbergen, 1979), the expert librarian is simply unable to cope with the onslaught of information that a computer can handle.

Learning systems are generally divided into three broad categories based on the way they are trained and used: supervised learning systems, unsupervised learning systems, and reinforcement learning systems (Mitchell, 1997). Whatever the characteristics of learning systems, however, all suffer from one problem that plagues computer systems: intractability. With learning systems, there are two major parameters of complexity leading to intractable behaviour: the number of attributes in an application domain, namely *dimensionality*, and the number of examples in a dataset. The latter typically applies only to the training stage of the system and, depending on intended use, may be acceptable. Data dimensionality, on the other hand, is an obstacle for both the training and runtime phases of a learning system. Many systems exhibit non-polynomial complexity with respect to dimensionality, which imposes a ceiling on the applicability of such approaches, especially to real world applications, where the exact parameters of a relation are not necessarily known, and many more attributes than needed are used to ensure all the necessary information is present. The curse of dimensionality effectively limits the applicability of learning systems to small, well-analysed domains, rendering otherwise elegant methodologies incapable of performing satisfactorily on arbitrary domains.

Rough set theory (Pawlak, 1991) is a formal methodology that can be employed to reduce the dimensionality of datasets as a preprocessing step to training a learning system on the data. Rough Set Attribute Reduction (RSAR) works by selecting the most information rich attributes in a dataset, without transforming the data, all the while attempting to lose no information needed for the classification task at hand (Chouchoulas and Shen, 1998; Shen and Chouchoulas, 2000). The approach is highly efficient, relying on simple set operations, which makes it suitable as a preprocessor for techniques that are more complex. Unlike statistical correlation-reducing approaches, RSAR requires no human input or intervention, or fine-tuning of parameters. The advantages of dimensionality reduction extend to the runtime of the system. By requiring fewer observations per datum, the reduced dimensionality learning system becomes more compact and its response time decreases. The cost of obtaining data drops accordingly, as fewer connections to instrumentation need be maintained. Finally, the overall robustness of the system increases, since, with fewer instruments, the chances of instrumentation malfunctions leading to spurious readings are reduced dramatically. RSAR also retains the semantics of the data, which makes the technique more transparent to human scrutiny, while it enhances any systems that benefit from semantics, such as fuzzy systems (Zadeh, 1975).

This paper investigates the application of RSAR to both supervised and unsupervised learning, in an effort to produce a generic, flexible framework for dimensionality reduction. Two separate systems are built, using supervised and unsupervised learning, respectively. Lozowski's fuzzy Rule Induction Algorithm (RIA) (Lozowski *et al.*, 1996) is used as an example of a supervised learning system. Friedman's Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) are employed to represent

unsupervised learning systems. To gauge the success of the two integrated systems, they are used to build a model of river algae growth as influenced by changes in the concentration of several chemicals in the water. The success of the application is demonstrated by the reduction in the number of measurements required, in tandem with accuracy that matches very closely that produced by training on the original, unreduced dataset.

The paper describes rough set theory and RSAR, as well as the chosen representatives of supervised and unsupervised learning. The algae application domain and its adaptation for use by the two systems are discussed, followed by detailed experimental results showing how RSAR manages to reduce dimensionality by losing as little information as possible.

2. Background

2.1. Rough Set Theory and RSAR

Rough set theory (Pawlak, 1991) is a formal mathematical tool that can be applied to reducing the dimensionality of datasets. The rough set attribute reduction (RSAR) method removes redundant input attributes from datasets of discrete values, all the while making sure that no information is lost. The approach is fast and efficient, making use of standard operations from conventional set theory.

To demonstrate the RSAR algorithm, an example will be followed through. Suppose that a dataset $\mathbb{D}$ is viewed as a table, where attributes are columns and objects are rows, as in Table 1 (adapted from (Pawlak, 1991)). Let U denote the set of all objects in the dataset, A the set of all attributes, C the set of input attributes, and D the set of output attributes. Thus, in this example, $U = \{0, 1, 2, 3, 4, 5, 6, 7\}$, $A = \{a, b, c, d, e\}$, $C = \{a, b, c, d\}$, and $D = \{e\}$.

Table 1. An example dataset.

$x \in U$	a	b	c	d	e
0	1	0	2	2	0
1	0	1	1	1	2
2	2	0	0	1	1
3	1	1	0	2	2
4	1	0	2	0	1
5	2	2	0	1	1
6	2	1	1	1	2
7	0	1	1	0	1

The value of attribute $q \in A$ in object $x \in U$ is written as $f(x, q)$, which defines an equivalence relationship over U. With respect to a given q, the function partitions the universe into a set of pairwise disjoint subsets of U

$$R_q = \{x\colon \ x \in U \ \ \wedge \ \ f(x, q) = f(x_0, q) \ \ \forall \ \ x_0 \in U\}.$$

For instance, for the discussed example, $R_a = \{\{1,7\},\{0,3,4\},\{2,5,6\}\}$.

Consider a subset of the set of attributes, $P \subset A$. Two objects numbered x and y in U are *indiscernible* with respect to P if and only if $f(x,q) = f(y,q)\ \forall\ q \in P$. The indiscernibility relation for all $P \in A$ is written as $IND\,(P)$. $U/IND\,(P)$ is used to denote the partition of U given $IND\,(P)$ and is calculated as

$$U/IND\,(P) = \bigotimes \Big\{ q \in P\colon\ \ U/IND\,(q) \Big\},$$

where

$$A \otimes B = \{X \cap Y\colon\ \ \forall\ X \in A,\ \forall\ Y \in B,\ X \cap Y \neq \emptyset\}.$$

For instance, if $P = \{b,c\}$, objects 0 and 4 are indiscernible; 1, 6 and 7 likewise. The rest of the objects are not. This applies to the example dataset as follows:

$$\begin{aligned} U/IND\,(P) &= U/IND\,(b) \otimes \mathrm{U/IND}(c) \\ &= \Big\{\{0,2,4\},\{1,3,6,7\},\{5\}\Big\} \otimes \Big\{\{2,3,5\},\{1,6,7\},\{0,4\}\Big\} \\ &= \Big\{\{2\},\{0,4\},\{3\},\{1,6,7\},\{5\}\Big\}. \end{aligned}$$

If $P = \{a,b,c\}$, then, similarly $U/IND\,(P) = U/IND\,(a) \otimes U/IND\,(b) \otimes U/IND\,(c)$.

Rough sets approximate traditional sets by using a pair of sets, named the *lower* and *upper approximations* of the set in question. The lower and upper approximations of a set $P \subseteq U$ (given an equivalence relation $IND\,(P)$) are defined as $\underline{P}Y = \bigcup\{X\colon\ X \in U/IND\,(P), X \subseteq Y\}$ and $\overline{P}Y = \bigcup\{X\colon\ X \in U/IND\,(P), X \cap Y \neq \emptyset\}$, respectively. Assuming that P and Q are equivalence relations in U, the *positive region* $POS_P(Q)$ is defined as $POS_P(Q) = \bigcup_{X \in Q} \underline{P}X$. A positive region contains all the objects in U that can be classified into attribute set Q using the information in attribute set P. For example, assuming $P = \{b,c\}$ and $Q = \{e\}$, $POS_{IND(P)}(IND\,(Q)) = \bigcup\{\{\},\{2,5\},\{3\}\} = \{2,3,5\}$.

What this means is that, with respect to input attributes b and c, objects 2, 3 and 5 can definitely be classified in terms of output attribute e. The remaining objects could, possibly, be classified, but this is not certain. The *degree of dependency* of a set Q of output attributes on a set of input attributes P is defined as $\gamma_P(Q) = \|POS_P(Q)\| \times \|U\|^{-1}$, where $\|A\|$ denotes the cardinality of set A. The complement of γ gives a measure of the contradictions in the selected subset of the dataset. If $\gamma = 0$, there is no dependence; for $0 < \gamma < 1$, there is a partial dependence. If $\gamma = 1$, there is complete dependence.

It is now possible to define the *significance* of an attribute. This is done by calculating the change of dependency when removing the attribute from the set of considered input attributes. Given P, Q and an attribute $x \in P$, $\sigma_P(Q,x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q)$. The higher the change in dependency, the more significant x is. This allows the calculation of the significance of any input attribute, for instance a as $\sigma_P(Q,a) = \gamma_{\{b,c\}}(\{e\}) = 1/8$.

This shows that attribute a is not indispensable, having a significance of 0.125, while attributes b and c can be dispensed with, as they do not provide any information that is significant for the classification of the data objects into the class values in e.

Attribute reduction involves the removal of attributes that have no significance to the classification at hand. An *attribute reduct set* (or simply *reduct*) is then defined as a subset R of the set of input attributes C such that $\gamma_C(D) = \gamma_R(D)$. For the set of output attributes D, it is obvious that a dataset may have more than one attribute reduct set. The set $\mathcal{R}$ of all attribute reduct sets R is defined as $\mathcal{R} = \{X : X \subseteq C, \ \gamma_C(D) = \gamma_X(D)\}$. The RSAR will not compromise with a set of input attributes that has a large part of the information embedded in the initial input attribute set, C—it *always* attempts to reduce the attribute set while losing *no* information that is significant to the classification at hand. RSAR searches for the attribute reduct sets of least cardinality. That is, it seeks one or more elements in the set of *minimal reducts* $\mathcal{R}_{\min} \subseteq \mathcal{R}$, where $\mathcal{R}_{\min} = \{X : X \in \mathcal{R}, \ \forall Y \in \mathcal{R}, \ \|X\| \leq \|Y\|\}$.

In terms of computational complexity and memory requirements, the calculation of all possible subsets of a given set is an NP-hard task. To solve this problem, the reduct subset search space is treated as a tree traversal. Each node of the tree represents the addition of one input attribute to an initially empty reduct. Instead of generating the whole tree and picking the best path on it, the path is chosen progressively. Starting with the empty set, attributes are chosen and progressively added to the candidate reduct until a $\gamma_P(Q)$ of 1 is reached, when all attributes have been added, or when the addition of an attribute does not change the value of γ. Attributes are added using the following heuristic: the next attribute chosen to be added to the candidate reduct is the attribute that adds the most to the reduct's dependency. Adding all attributes may not necessarily result in a γ of 1, in which case the dataset could not be correctly classified to begin with. This is dubbed the QuickReduct, also described in (Shen and Chouchoulas, 2000). QuickReduct is similar to the algorithm introduced in (Jelonek *et al.*, 1995), where it was proposed in conjunction with neural network-based classifiers.

It is now possible to show the workings of QuickReduct in the context of an example. The reduct R starts off as the empty set. For each of the attributes, γ is calculated with respect to the output attribute e: $\gamma_{\{a\}}(e) = 0/8$, $\gamma_{\{b\}}(e) = 1/8$, $\gamma_{\{c\}}(e) = 0/8$, and $\gamma_{\{d\}}(e) = 2/8$. This shows that attributes a and c are not of much use on their own. Attribute d contributes the most information, allowing the classification of two of the eight examples. QuickReduct hence adds d to the reduct and attempts to evaluate the addition of a second attribute from those remaining (a, b, c): $\gamma_{\{d,a\}}(e) = 3/8$, $\gamma_{\{d,b\}}(e) = 8/8$, and $\gamma_{\{d,c\}}(e) = 8/8$. The addition of either attribute b or c to the reduct (currently $\{d\}$) allows perfect classification of the data. The first such attribute, b, is added to the reduct.

RSAR offers a number of advantages. It preprocesses datasets without altering the attribute values themselves, thus maintaining the semantics. It is not a lossy algorithm; it will remove an input attribute from the dataset only if this action removes absolutely no information (with respect to the classification at hand). Unlike statistical correlation-reducing approaches like the Principal Components Analysis (PCA),

discussed among other places in (Haykin, 1994), the dimensionality reduction does not require a human input, or the setting of variance thresholds. The same feature is also a disadvantage, since other techniques offer a more aggressive dimensionality reduction, accepting that in some cases the loss of a little information may in fact prove to be advantageous (e.g. in noisy environments). However, it is possible to obtain a 'compromise' reduct by setting a threshold t for the degree of dependency. Then, QUICKREDUCT terminates when γ reaches t, instead of 1, producing a reduct with a certain loss of information, as specified by t. Another disadvantage of the RSAR algorithm is its lacking efficiency. This is successfully rectified by the QUICKREDUCT algorithm, which converts an exhaustive evaluation of all attribute combinations into a best-first tree search. In terms of the input domain, RSAR is mainly intended for discrete domains, although it can be adapted to cope with continuous ones, as described in (Shen and Chouchoulas, 2000).

2.2. Supervised Learning

Supervised learning involves learning from input-output pairs using inductive methodologies. A pre-labelled dataset is required to train such a learning algorithm. The dataset provides the learning system with a class of functions and a number of sample points for each function. Training involves approximating the functions in question. Members of this wide family of systems learn to classify data into a number of pre-defined classes or clusters.

Supervised learning is of particular use when systems under training are intended to perform tasks that have previously been performed by humans with a certain degree of success. In such cases a relation between data is known to exist, but the rules governing it are not known, or are difficult to obtain. The system to be trained effectively learns by example, generalising the knowledge to apply it to the entire domain.

Most connectionist approaches are supervised learning systems (Ripley, 1996). Typical approaches to rule induction also depend on supervised learning. The following is a description of the rule induction algorithm used to demonstrate the application of RSAR to this type of learning.

Lozowski's Rule Induction Algorithm. The rule induction algorithm presented in (Lozowski *et al.*, 1996) extracts fuzzy rules from real-valued examples. Although this data-driven RIA was proposed to be used in conjunction with neural network-based classifiers, it is independent of the type of the classifier used (Shen and Chouchoulas, 1999). Provided with training data, the RIA induces approximate relationships between the characteristics of the conditional attributes and those of the decision attributes. The conditional attributes of the induced rules are represented by fuzzy variables, facilitating the modelling of the inherent uncertainty of the application domain.

The algorithm generates a hyperplane of candidate fuzzy rules ($p_1 \wedge p_2 \wedge \cdots \wedge p_n \Rightarrow c$) by fuzzifying the entire dataset using all the combinations of rule conditions. Thus, a domain with n conditional attributes, each of which is a fuzzy region fuzzified by f_x

fuzzy sets ($1 \leq x \leq n$), the hyperplane is fuzzified into $\prod_{i=1}^{n} f_i$ n-dimensional clusters, each representing one vector of rule conditions. Each cluster $\mathbf{p} = \langle \mu_1, \mu_2, \ldots, \mu_n \rangle$ may lead to a fuzzy rule, provided that training examples support it. To obtain a measure of what classification applies to a cluster, fuzzy min-max composition is used. The conditional attribute values of each training example are fuzzified according to the fuzzy conditions $\langle \mu_1, \mu_2, \ldots, \mu_n \rangle$ that make up cluster $\mathbf{p}$. For each example $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$, $S_c^{\mathbf{p}}\mathbf{x} = \min(\mu_1(x_1), \mu_2(x_2), \ldots, \mu_n(x_n))$ is calculated. This is the s-norm of example $\mathbf{x}$ with respect to cluster $\mathbf{p}$ and classification c. To give a measure of the applicability of a classification to cluster $\mathbf{p}$, the maximum of all s-norms with respect to $\mathbf{p}$ and c is calculated (this is dubbed a t-norm): $T_c^{\mathbf{p}} = \max\{S_c^{\mathbf{p}}\mathbf{x} \ \forall \ \mathbf{x} \in D_c\}$, where D_c is the set of all dataset examples that can be classified as c. This is iterated over all possible classifications c to provide a full indication of how well each cluster applies to each classification.

A cluster generates at most one rule. The rule's conditions are the cluster's n coordinate fuzzy sets. The conclusion is the classification attached to the cluster. Since there may be t-norms for more than one classification, it is necessary to decide on one classification for each of the clusters. Such contradictions are resolved by using the *uncertainty margin*, ε ($0 \leq \varepsilon < 1$). This means that a t-norm assigns its classification on its cluster if and only if it is greater by at least ε than all the other t-norms for that cluster. If this is not the case, the cluster is considered undecidable and no rule is generated. The uncertainty margin introduces a trade-off to the rule generation process. In general, the higher ε is, the fewer rules are generated, but the classification error may increase.

Lozowski's RIA is NP-hard, and may become intractable when inducing rules for datasets with many conditional attributes (Chouchoulas and Shen, 1998). The most important problem, in terms of both memory and runtime, is dealing with the large numbers of combinations of fuzzy values. This is not so important when only a few attributes are involved. Applied to a more complex problem, such as algae population estimation (see Section 3), without some means of attribute reduction, the algorithm's intractable nature becomes evident, both in terms of time and space.

It is thus convenient and helpful to treat the creation of fuzzy-set vectors as the creation of a tree. In this context, a leaf node is one combination of membership functions, and each arc represents one evaluation of a membership function. The minimum membership is retained when creating the t-norms. Any membership function that evaluates to zero means that all leaf nodes in the subtree will eventually evaluate to zero, too, because of the use of the $\min(\cdot)$ function. A subtree is therefore useless and can be pruned if (and only if) its root node evaluates to zero.

In an application domain where a reasonable degree of resolution is required, it is not unusual to see quantities partitioned into five or seven fuzzy sets. Assuming an average of six fuzzy sets per attribute and 40 attributes, the data may be seen as a 40-dimensional ($\mathbb{R}^{40}$) hyperplane, each dimension of which is a fuzzy region covered by six fuzzy sets. The RIA would attempt to produce rules to cover the entire space by using each fuzzy set of each dimension. Thus, it would need to generate at most 6^{40} possible rules.

In most applications of fuzzy logic, any given value x in a fuzzy region will belong to *at most* two fuzzy sets, A and B, with membership $\mu_A(x) \geq \mu_B(x) > 0$. Thus, for any other fuzzy set F_i, it may be assumed that $\mu_{F_i}(x) = 0$.

The RIA pruning algorithm will detect this at an early stage and will not consider fuzzy sets F_i as containing candidate rules. Therefore, for each of the 40 fuzzy regions (dimensions of the hyperplane), two of the six fuzzy sets will be allowed to generate candidate rules. This reduces the number of combinations to at *worst* 2^{40}. If some values in a fuzzy region only belong to *one* fuzzy set with non-zero membership, this number becomes smaller.

Even given the worst case scenario, however, the time needed by the enhanced algorithm for this example is approximately nineteen orders of magnitude less than that needed for the full tree traversal. The savings are significant, but the number of combinations is still far too large.

2.3. Unsupervised Learning

Unsupervised learning systems discover patterns within the data. There are no pre-defined classes, hence the learning system is self-organising. Since such systems need no pre-labelled data, they can be applied to domains where labelling is difficult for humans to perform, or simply unknown. Unsupervised learning systems can perform clustering to discover new relations between data, based on an internal quality measure.

Thus, such systems are of particular use where the relations governing a domain are yet to be discovered. Alternatively, unsupervised learning can be used where a domain contains samples of different, unknown classes. Here, the learning system discovers relations within the data without the benefit of pre-defined classes or clusters. New domains, those that have heretofore not been tackled by humans, or those where human expertise is in question (or needs to be improved by the use of the system), are especially suitable for unsupervised learning.

Clustering approaches are typical examples of unsupervised learning. The following section describes the unsupervised algorithm used to demonstrate the application of RSAR to this type of learning systems.

Multivariate Adaptive Regression Splines. Friedman's MARS (Friedman, 1991) is a statistical methodology that can be trained in an unsupervised manner to approximate multidimensional functions. It uses recursive partitioning and spline curves to closely approximate the underlying problem domain. The partitioning and number of basis functions used are automatically determined by this approach based on the provided training data.

A spline is a parametric curve defined in terms of control points, also referred to as knots, and a basis function or matrix (Foley *et al.*, 1990). The curve approximates the line joining the knots, as shown in Fig. 1. Each knot has an associated weight. The spline is a continuous curve that approaches its control points. Although splines generally do not interpolate their control points, they can approximate them

quite closely. Increasing a control point's weight makes the spline come closer to the point. The basis function or matrix provides the spline with its characteristics. Two- and three-dimensional splines are widely used in computer graphics and typography (Bartels *et al.*, 1987).

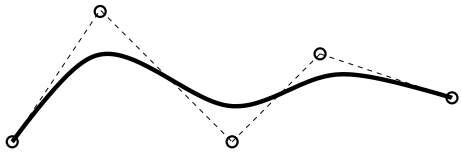

Fig. 1. A spline curve and its control points.

MARS adapts the general, n-dimensional form of splines for function approximation. It generates a multi-dimensional spline to approximate the shape of the underlying problem hyper-plane. Each attribute is recursively split into regions and subregions. The split into subregions is performed if a spline cannot approximate a region within reasonable bounds. A hierarchy of spline basis functions is thus built. This allows MARS for great flexibility and autonomy in approximating numerous deceptive functions.

MARS models may be expressed in the following form, known as ANOVA decomposition (Friedman, 1991):

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) + \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k) + \cdots ,$$

where $\mathbf{x}$ is an input vector whose ordinates represent a training or testing datum, a_0 is the coefficient of the constant basis function B_1 (Friedman, 1991), f_i is a univariate basis function of x_i, f_{ij} is a bivariate basis function of x_i and x_j, and so on. In this context, K_m is the number of variables a basis function involves. The ANOVA decomposition shows how a MARS model is the sum of basis functions, each of which expressing a relation between a subset of the variables of the entire model. As an example, a univariate basis function f_i is defined as $f_i(x_i) = \sum_{K_m=1} a_m B_m(x_i)$, where a_m is the coefficient of basis function B_m which only involves variable x_i. In turn, B_m is defined as $B_m(\mathbf{x}) = I[\mathbf{x} \in R_m]$, where I is a Boolean function that evaluates to 1 if $\mathbf{x}$ is a point within a predefined region R_m, and evaluates to 0 otherwise. Regions R_i may overlap, so there can be no zero pairwise product expectation. MARS uses a generalised, multivariate spline basis function $B_m^q(\mathbf{x}) = \prod_{k=1}^{K_m} [s_{km}(x_{km} - t_{km})]^q$, where B_m is the basis function in question, involving K_m ordinates x_{km} of point $\mathbf{x}$ ($1 \leq k \leq K_m$); q is the order of the multivariate spline, with $q \geq 1$; $s_{km} = \pm 1$; and t_{km} is ordinate m of the spline knot $\mathbf{t}_m$.

Recursive partitioning is employed in MARS in order to perform two tasks: to adjust the basis functions' coefficients ($\{a_m\}_1^M$, $1 \leq m \leq M$, for each basis function B_m, where M is the number of basis functions generated by the algorithm as a result of recursive partitioning), and to partition the universe of discourse into a set of these

disjoint regions $\{R_m\}_1^M$. A region R is split into two subregions if and only if a basis function cannot be adjusted to fit the data in R within a predefined margin. Recursive partitioning is discussed in detail in (Friedman, 1991).

Unlike many other function approximators, this system produces continuous, differentiable approximations of multidimensional functions, thanks to the use of splines. Applied to most domains, MARS is particularly efficient and produces good results. The continuity of the resultant approximative models is one of the most desirable results if statistical analysis is to be performed. However, MARS suffers from the curse-of-dimensionality problem, especially when dealing with complex domains. Each dimension of the hyperplane requires one dimension in the approximation model, and an increase in the time and space required to compute and store the splines. The time required to perform predictions increases exponentially with the number of dimensions. Further, MARS is very sensitive to outliers. Noise may mar the model by causing MARS to generate a much more complex model as it tries to incorporate the noisy data into its approximation. A technique that simplified the produced models and did away with some of the noise would thus be very desirable. This forms the very reason that the Rough Set-Based Attribute Reduction technique is adopted herein to build an integrated approach to multivariate regression with reduced dimensionality.

3. Problem Domain

Awareness of environmental issues has increased greatly in recent years. Waste production from a vast number of manufacturing processes is one of the most important issues, as it influences algae[1] growth patterns in rivers. Growing algae communities are detrimental to water clarity, while complex water life like fish can also be endangered, due to changes in the oxygen content of the water. Human activities can also be affected, since toxic effects may be present in relation to algae growth. Measuring and reducing the impact that farming, manufacturing and waste disposal have on nutrient content in rivers have, thus, attracted much attention. Biologists are attempting to locate the chemical parameters that control the growth of algae communities (ERUDIT, 1999).

To help in this task, an intelligent tool would be desirable. The system should locate the parameters that control algae population fluctuations and use this information to estimate these changes. Such a system could aid in a number of areas, including simulating hypothetical scenarios and predicting trends in algae communities, in addition to its intended estimation task.

To build the knowledge base for this application, samples from different European rivers were taken over the period of one year. These samples were analysed to quantify the presence of several chemicals, including nitrates, nitrites and ammonia, phosphate, oxygen and chloride. The pH of the water was also measured. In addition, the algae population distributions for each of the species involved were determined

[1] The alga is a single-celled plant that has, over a period of three and a half billion years, evolved into the most successful coloniser of almost any known ecology on the planet.

in the samples. A number of additional factors were taken into account, such as the season, river size and flow rate.

It is relatively easy to locate relations between one or two of these quantities and a species of algae. However, the process involves expertise in chemistry and biology and requires well-trained personnel and microscopic examination that cannot be automated given the state of the art. Thus, the process becomes expensive and slow, even for a subset of the quantities involved here. There are complex relations at work between the attributes of this application domain: algae may influence one another, as well as be influenced by the concentration of chemicals. As such, there is expected to be some redundancy in the data, allowing for a good case study of RSAR.

The dataset (ERUDIT, 1999) available for training includes 200 instances. Each instance contains the following information: the time of year the sample was taken, given as a season; river size; water flow rate; eight chemical concentrations; and population counts for seven algae species. The first three attributes of each instance (season, river size and flow rate) are represented as fuzzy linguistic variables. Chemical concentrations and algae population estimates are represented as continuous quantities. The dataset includes a few samples with missing values. Of the 200 instances, two exhibiting mostly unknown values were removed from the data because of their low quality.

In order for the RIA to generalise given training samples, attributes of numerical values are fuzzified. As the first three attributes are already represented in fuzzy terms, no such preprocessing is required for them. In the case of MARS, the first three attributes were defuzzified into discrete integers. Since MARS deals with numerical values, no preprocessing was needed for the remaining attributes.

Matters differ for the eight chemical concentrations. As with all concentrations, these exhibit an exponential distribution. The nature of the samples is such that there are not enough representative values in a homogeneous distribution in the attributes' domains. Thus, samples are described using a logarithmic scale defined by $f(x) = \log(x + 1)$, where x is the numerical measurement of an attribute[2].

As can be expected, the distributions of the algae are also exponential. This, coupled with the fact that the decision attributes representing algae population counts are numerical, suggests the use of a similar treatment as above. The conditional attributes were thus transformed by $g(x) = \lfloor \log(x + 1) \rfloor$, where x is the numerical measurement of the algae community's population. This quantisation is required because RSAR works better with discrete classes. The quantised data are only used for dimensionality reduction. During training, the original data are employed. The data were still expressed on a logarithmic scale using $g_0(x) = \log(x + 1)$, but without the quantisation introduced by the use of the floor function ($\lfloor \cdot \rfloor$).

This is reasonable because of the way the algae population 'counts' are obtained. It is assumed that the river's water is perfectly homogeneous and that any sample of the water, no matter how small, is statistically representative. Water samples are thus obtained. A few drops of each sample are examined visually via microscope

[2] Concentrations are non-negative real numbers, hence it is necessary to add an arbitrary constant to avoid the logarithm of zero.

and the number of algae are counted. This allows for human errors to determine the population, as well as the fact that a number of drops of water from a sample of a river are not necessarily statistically representative of the entire river. Quantisation alleviates this problem. In addition, if the aim is to estimate the behaviour of algae communities, it is far more intuitive to provide linguistic descriptions like 'normal', 'lower' and 'higher' rather than estimated concentrations that have to be matched against tables and may again be subject to human error.

4. Utility of RSAR in Learning

4.1. System Integration

In essence, the approach proposed herein deals with large datasets by applying RSAR to the dataset to discover a set of attributes that convey all the information with as little redundancy as possible. The desired attributes are then extracted from the dataset and fed to either the RIA to induce a suitable ruleset, or to MARS to regress a multivariate spline model of the domain.

As shown in Fig. 2, the system integrates the following modules:

Precategorisation reads a dataset and outputs a version in which continuous values have been replaced by discrete labels. A standard fuzzifier (Chouchoulas and Shen, 1998) may be employed for this task.

Attribute Reduction implements the RSAR algorithm, as described in Section 2.1.

Attribute Selection is a trivial sub-program that, given a set of attributes (by the attribute reduction module) and a dataset, extracts and outputs only the specified attributes and their real values from the dataset.

Knowledge Induction incorporates either the MARS (described in Section 2.3) or RIA (see Section 2.2) techniques, depending on the supervised or unsupervised nature of the task at hand. MARS generates a spline model of the domain, whereas the RIA produces a fuzzy ruleset.

The effectiveness and efficiency of the integrated approach are demonstrated in the following two sections, which provide results for the supervised (FuREAP) and unsupervised (RSAR+MARS) applications, respectively.

For convenience, each of the seven alga species was processed separately by the learning systems in order to provide seven different rulesets or models. Each ruleset or model reflects the behaviour of one species. The separate RIA rulesets can be merged trivially, to form a single ruleset. Alternatively, the RIA can be applied to all seven to produce directly a single, unified ruleset. This latter choice is, of course, a more inelegant and inflexible solution than having separate algae models. Therefore, the following results are shown with respect to individual algae species. Please note that MARS only approximates functions with a single output, so splitting the domain into seven sub-problems, one for each alga species, is the only way to use this dataset with MARS.

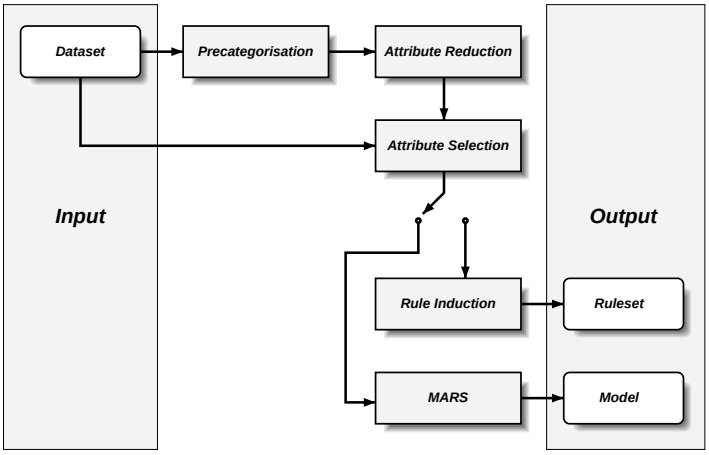

Fig. 2. Block diagram of the integrated system.

4.2. Supervised Learning (FuREAP)

It is, first of all, interesting to investigate what effects dimensionality reduction may have on the runtime performance of the Fuzzy-Rough Estimator of Algae Populations (FuREAP). To show whether it has an impact on the overall accuracy, Lozowski's RIA algorithm was used to induce a ruleset from the entire, unreduced algae dataset. The results are shown in the top row of Fig. 3. Then, FuREAP was instructed to reduce the dimensionality of the dataset and produce another ruleset from these reduced data. This resulted in a seven-attribute dataset selected from the original, eleven-attribute one. The results of testing this ruleset are illustrated in the bottom row of Fig. 3.

Experimental results are given as two types of graphs: estimation error and ruleset size. Both quantities are plotted against ϵ, the uncertainty margin or tolerance which creates a trade-off between the estimation accuracy of the ruleset and the number of learned rules it comprises. Please note that the estimation error, rather than the estimation accuracy, is shown here. This is done to emphasise the accuracy/size trade-off. The ruleset size grows exponentially, so the graphs involving it are shown on a logarithmic scale. All seven algae species are shown separately on each graph as a family of curves. Also, please note that, in plotting the graphs, 'undecidable' answers by the RIA were considered wrong answers, thus giving slightly more conservative results.

There is a certain drop in accuracy (around 10%) after dimensionality reduction, which may indicate that the attribute reduction process has removed some of the necessary information. However, a full investigation of the domain reveals that inexpert fuzzification is largely responsible for the error during the rule-induction phase. The fuzzification of certain conditional attributes is less successful than others. This causes the removal of some of the better-fuzzified attributes during dimensionality reduction, leading to the observed drop in accuracy.

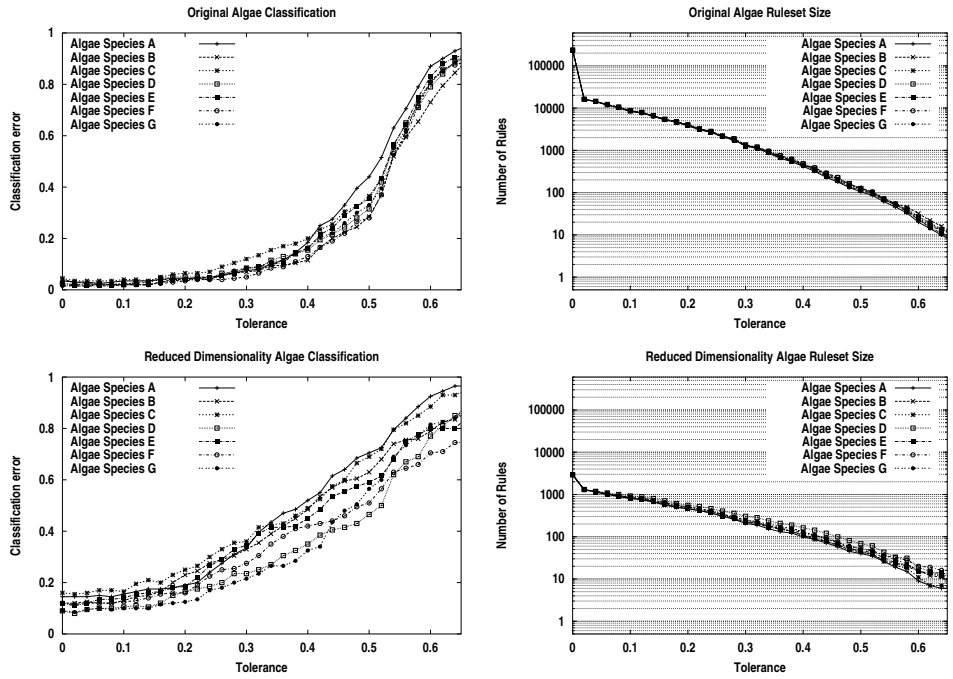

Fig. 3. Algae estimation accuracy before (top) and after (bottom) dimensionality reduction. The left graphs show estimation *error* against the value of ε; the right graphs show the ruleset size (on a logarithmic scale) against ε.

Despite this accuracy reduction, however, the ruleset induced from the low-dimensionality data is around two orders of magnitude smaller than that generated from the unreduced dataset. Induction speed increases at a higher rate, making a strong argument for the use of FuREAP in applications where time and storage are at a premium. As stated previously, however, the speed and storage benefits are not limited to the training stage. They extend to the runtime use of the system. By reducing the dimensionality of the dataset, the arity of the rules is also decreased. This allows for fewer measured variables, which is important for dynamic systems where observables are often restricted, or where the cost of obtaining more measurements is high. In the river algae domain, for instance, providing different measurements has different costs attached. It is trivial to give the time of year and size of river, but the flow rate may need extra equipment. Additionally, each of the measurements of concentration of chemicals may need its own process, requiring time, well-trained personnel and money. Reducing the number of measurements to be made significantly enhances the potential of the estimator system.

To show that the dimensionality reduction part of FuREAP performs as claimed, it is desirable to prove two further points: that the RSAR algorithm in FuREAP truly finds the smallest, best subset of conditional attributes of the dataset (known as a reduct), and that adding further attributes to this reduct does not produce better results.

To this end, two further experiments were conducted. In the first one, numerous datasets of six attributes each were randomly generated from the original, eleven-attribute algae dataset. Rulesets were induced from these, and the average estimation error of all runs was plotted, as shown on the right graph of Fig. 4 (where the left graph is the reduced dataset error, copied here to ease comparison). Two empirical conclusions can be drawn from these results: first, not all attributes contribute the same information; second, the results obtained from random sets of attributes are worse than those obtained from the reduct set. The latter conclusion demonstrates that RSAR does indeed locate the minimal high-quality attribute set.

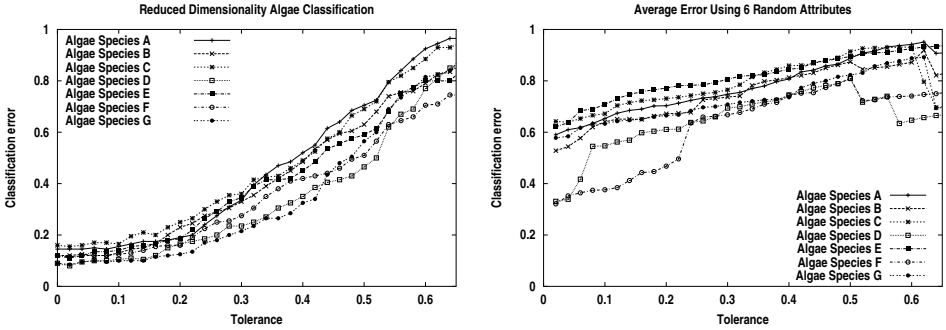

Fig. 4. Comparison of the estimation error after training on the reduct set of seven attributes (left), and random sets of six attributes (right).

In the second further experiment, the four remaining conditional attributes were added to the seven-attribute reduct one at a time. The aim was to show that more attributes do not necessarily imply higher accuracy. Rulesets were induced from these artificially produced attribute sets, and the results were averaged. As shown on the right graph of Fig. 5 (again, the canonical, reduced results are shown on the left graph for comparison), error increased by adding an arbitrary attribute to the reduct. This leads to the conclusion that the reduct indeed leads to the minimal accuracy loss.

It is clear that FuREAP performs very well. This shows that real-world problems do contain a lot of redundancy which, once removed, allows highly accurate rulesets of low-arity rules to be induced. To reinforce the significance of the present approach, the performance of FuREAP is compared with that of a system employing rules generated using C4.5, the standard machine learning tool (Quinlan, 1993), from the sample dataset. FuREAP is able to provide an estimation accuracy that surpasses that of C4.5, all the while using a smaller set of conditional attributes (as shown in Table 2). Although C4.5 offers superior training speed, the number of attributes involved in the final system is very important, inasmuch as the cost, complexity and time requirements of obtaining each set of measurements is proportional to the number of measurements in each set.

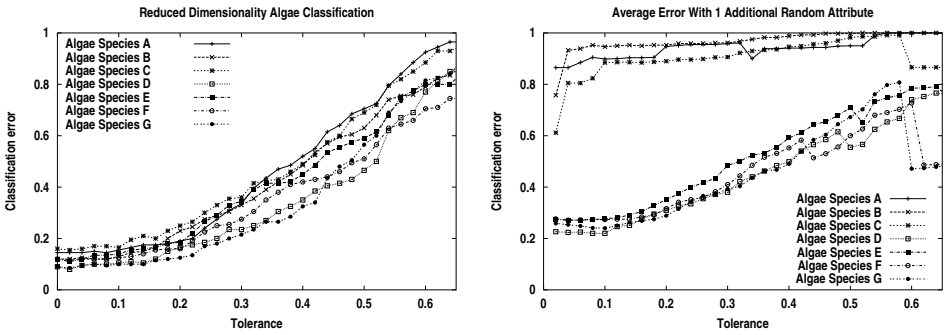

Fig. 5. Comparison of the estimation error after training on the reduct set of attributes (left), and the reduct set plus one random attribute (right).

Table 2. Comparison between FuREAP and C4.5 with respect to accuracies and the number of conditional attributes involved.

Algae	FuREAP		C4.5	
Species	Error	Attributes	Error	Attributes
Species A	15%	7	18%	11
Species B	12%	7	19%	10
Species C	16%	7	24%	9
Species D	8%	7	13%	11
Species E	11%	7	14%	10
Species F	12%	7	15%	10
Species G	9%	7	16%	11

4.3. Unsupervised Learning (RSAR+MARS)

To test how unsupervised learning systems can benefit from the use of RSAR, two series of experiments were performed: one produced MARS models based on the original, unreduced algae data; the other employed RSAR to reduce the dimensionality of the data and invoked MARS to produce models. The algae dataset was split randomly (using a 50% split ratio) into training and test datasets, both 'massaged' as described earlier using a 50% split ratio. 100 runs were performed for each experiment series.

For the second experiment, RSAR was run on the suitably preprocessed algae dataset. The reduction algorithm selected seven of the eleven conditional attributes. This alludes to the fact that the dataset was reasonably information-rich before reduction, but not without redundancies.

The results are shown in Table 3. Minimum and maximum RMS errors are shown separately for each alga species. It is clear from these results that the implications of employing RSAR as a preprocessor for MARS are minimal. The slight drops in

accuracy exhibited after the dimensionality reduction indicate that the process has removed some of the necessary information. This information reduction was due to the quantisation process employed for this domain, rather than the RSAR methodology itself.

Table 3. Experimental results, showing RMS errors.

	Before		After	
Alga	Min	Max	Min	Max
Species A	0.923	1.639	0.924	1.642
Species B	0.893	1.362	0.932	1.389
Species C	0.822	1.202	0.856	1.206
Species D	0.497	0.748	0.595	0.768
Species E	0.723	1.210	0.768	1.219
Species F	0.762	1.158	0.892	1.259
Species G	0.669	0.869	0.689	0.872

However, MARS models obtained from the low-dimensionality data are smaller than their unreduced equivalents by at least a factor of 2^4. This is based on a conservative assumption that each of the four removed attributes is split into only two subregions by MARS. Given the relative complexity of even small MARS models, this reduction in the model size is particularly welcome. The processing time required by MARS decreases similarly, although the algorithm's efficiency is such that time requirements are not as important as space requirements.

As with FuREAP, reducing the dimensionality has welcome side effects extending to the runtime of the system: training time, runtime, response time and costs are reduced, while the speed and applicability of the system are increased.

5. Conclusion

Learning systems have found their way to all manners of application domains. This success is due to the fact that learning systems are cost-effective. The price of computing equipment has dropped dramatically over the past decade, while the time of human experts has remained steadily expensive. Having even a fraction of the knowledge of a highly paid and competent consultant built into an application system is clearly very desirable. This requires training or learning.

Regardless of whether supervised or unsupervised learning is used, however, many systems suffer from the same problem: intractability. Many systems exhibit non-polynomial complexity with respect to dimensionality, which imposes a ceiling on the applicability of such approaches, especially to real world applications, limiting the applicability of learning systems to small, well-analysed domains.

Rough set theory (Pawlak, 1991) is a formal methodology that can be employed to reduce the dimensionality of datasets as a preprocessing step to training a learning

system on the data. Rough Set Attribute Reduction (RSAR) works by selecting the most information rich attributes in a dataset, without transforming the data, all the while attempting to lose no information needed for the classification task at hand (Chouchoulas and Shen, 1998; Shen and Chouchoulas, 2000). The advantages of dimensionality reduction extend to the runtime of the system: systems become simpler, more compact and robust; response times drop; and costs related to obtaining data are reduced.

This paper has investigated the application of RSAR to both supervised and unsupervised learning, producing a flexible framework for dimensionality reduction. Two separate systems were built, using supervised and unsupervised learning, respectively. Lozowski's fuzzy Rule Induction Algorithm (RIA) (Lozowski *et al.*, 1996) was taken to represent supervised learning systems, while Friedman's Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) represented unsupervised learning. To gauge the success of the two integrated systems, they were applied to estimating river algae populations as influenced by changes in the concentration of chemicals in the water. The success of the application was evident by the reduction in the number of measurements required, as well as by the accuracy that matches closely that produced by training on the original, unreduced dataset.

References

Bartels R., Beatty J. and Barsky B. (1987): *Splines for Use in Computer Graphics and Geometric Modeling*. — Los Altos: Morgan Kaufmann.

Chouchoulas A. and Shen Q. (1998): *Rough set-aided rule induction for plant monitoring*. — Proc. Int. Joint Conf. *Information Science (JCIS'98)*, Research Triangle Park, NC, Vol.2, pp.316–319.

ERUDIT, European Network for Fuzzy Logic and Uncertainty Modeling in Information Technology. *Protecting Rivers and Streams by Monitoring Chemical Concentrations and Algae Communities (Third International Competition)* http://www.erudit.de/erudit/activities/ic-99/problem.htm

Foley J.D., van Dam A., Feiner S.K., Hughes J.F. and Philips R.L. (1990): *Introduction to Computer Graphics*. — Reading: Addison-Wesley.

Friedman J.H. (1991): *Multivariate adaptive regression splines*. — Annals of Statistics, Vol.19, No.1, pp1–67.

Haykin S. (1994): *Neural Networks*. — New York: Macmillan College Publ. Comp.

Jelonek J., Krawiec K. and Slowinski R. (1995): *Rough set reduction of attributes and their domains for neural networks*. — Comput. Intell., Vol.11, No.2, pp.339–347.

Lozowski A., Cholewo T.J. and Zurada J.M. (1996): *Crisp rule extraction from perceptron network classifiers*. — Proc. Int. Conf. *Neural Networks*, Washington, volume of plenary, panel and special sessions, pp.94–99.

Mitchell T.M. (1997): *Machine Learning*. — New York: McGraw-Hill.

Pawlak Z. (1991): *Rough Sets: Theoretical Aspects of Reasoning About Data*. — Dordrecht: Kluwer.

Quinlan J.R. (1993): *C4.5: Programs for Machine Learning*. — San Mateo: Morgan Kaufmann.

van Rijsbergen C.J. (1979): *Information Retrieval*. — London: Butterworths.

Ripley B.D. (1996): *Pattern Recognition and Neural Networks*. — Cambridge: Cambridge University Press.

Shen Q. and Chouchoulas A. (1999): *Data-driven fuzzy rule induction and its application to systems monitoring*. — Proc. 8-th IEEE Int. Conf. *Fuzzy Systems*, Seoul, Korea, Vol.2, pp.928–933.

Shen Q. and Chouchoulas A. (2000): *A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems*. — Eng. Appl. Artif. Intell., Vol.13, No.3, pp.263–278.

Zadeh L. (1975): *The concept of a linguistic variable and its application to approximate reasoning – I*. — Inform. Sci., Vol.8, No.1, pp.199–249.