Analysis and Evaluation of Learning Classifier Systems applied to Hyperspectral Image Classification

Arnaud Quirin, Jerzy Korczak
Université Louis Pasteur
LSIIT, CNRS
Strasbourg, France
{quirin,korczak}@lsiit.u-strasbg.fr

Martin V. Butz, David E. Goldberg
Illinois Genetic Algorithms Laboratory
University of Illinois
Urbana, IL, USA
{butz,deg}@illigal.ge.uiuc.edu

Abstract

In this article, two learning classifier systems based on evolutionary techniques are described to classify remote sensing images. Usually, these images contain voluminous, complex, and sometimes erroneous and noisy data. The first approach implements ICU, an evolutionary rule discovery system, generating simple and robust rules. The second approach applies the real-valued accuracy-based classification system XCSR. The two algorithms are detailed and validated on hyperspectral data.

1. Introduction

The emergence and the improvement of remote sensing, aircraft simulation, airborne and spaceborne sensor systems as well as other kinds of such survey technologies have enhanced our means to explore and to collect data considerably. However, this rapid increase in data results in more time and cost for storage as well as for the analysis and the mining of the data. At the same time, a lot of useless information can hide valuable information. These observations force data miners to focus on elaborated and sophisticated algorithms to overcome this rapid data growth. Strong effort has been devoted to develop new classification algorithms and improve techniques used to classify statistical data sets [**?**, **?**, **?**]. Relatively few data miners in the machine learning community have considered how classification rules might be discovered from raw and expertly classified images. The images, registered by various satellites (e.g. SPOT, CASI, Quick Bird), generally contain voluminous data. Sometimes they are very noisy due to the presence of various details in a high spatial resolution or unfavorable atmospheric conditions at the time the images were acquired. These data can embrace different cameras having various spectral and spatial resolutions [**?**].

It appears that learning classifier systems (LCS) are well suited to remote image mining. Based on evolutionary algorithms [**?**], a LCS generate classification rules able to adapt themselves according to the available data, environment, and the evolution of classes. To our knowledge, there are only a few published studies on classifier systems applied to remote sensing images. This research focus on the *pixel classification problem* (one pixel is always classified in one class). ICU, developed at the LSIIT (software available on [**?**]), was compared to XCS, an other classifier system, known to be robust, according to many early studies [**?**, **?**, **?**]. ICU is an evolutionary rule discovery system combining a genetic algorithm (GA) and a population of classification rules describing constraints for each pixel from the data. XCS is a learning classifier system, developed by S. Wilson, that evolves a rule set online based on prediction accuracy and a niche genetic reproduction [**?**].

The paper is structured as follows. Basic terms and properties of hyperspectral images are introduced in section 2. Section 3 describes the two algorithms ICU and XCS. In this section, the main components and the quality measures of the two methods are explained. A CASI image is analyzed in section 4 and results are compared.

2. Hyperspectral remote sensing images

An hyperspectral image is a set of two dimensional arrays $I_{X,Y,S}$ where (X, Y) is respectively the width and the height of the image and S the number of spectral channels (or spectral bands). The term *hyperspectral* refers to an image which includes more than 20 spectral bands (similar to those produced by ROSIS and DAIS). Conversely, the term *multispectral* is used in the case of a low number of spectral bands, as CASI and Quick Bird remote sensors. A value $I(x, y, s)$ in this array is the reflectance observed on the

pixel location (x, y) at the wavelength corresponding to the spectral channel s. A reflectance value corresponds to the intensity of the response obtained from the ground.

To illustrate our approach, a multispectral image from the *CASI* sensor has been used (Airborne spectrometer, 1175x673, 15 spectral channels (0.43 μm - 0.87 μm), high resolution (1.3m)). Learning and testing were applied on subsets of this image (142x99 points), and only 1540 points were validated by human ground truthing. Thus, according to the validation strategies used (hold-one-out, cross-validation), testing sets represent 20% to 50% of the original validated image points.

3. Evolutionary methods to discover the classification rules

The idea of rule mining is to discover a pool of classifiers, where each classifier, taking a pixel as input, returns its class. The discovery of a classification rule is a combinatorial problem. The search space is very large. It depends on the size of the image, the pixel spectral resolution, and the rule encoding. Evolution-based approaches and particularly Learning Classifier Systems (LCS, [**?**]) are specifically designed to search efficiently the space of classifiers.

For a long time, many generic classifiers systems were developed to explore the potential of these techniques including ZCS, LFCS [**?**], S-classifiers, ICU [**?**], XCS-R [**?**] and Fuzzy XCS. XCS was chosen to compare with ICU because it is one of the most used and developed algorithm in the field of learning classifier systems. The key differences between the two systems are the following: the fitness (based on predictive positive accuracy in ICU, on accuracy in XCS), the rule representation (conjunctions of disjunctions in ICU, disjunctions in XCS) and the population (only one rule for each class in ICU, a whole population set in XCS). The classical binary strings of classifier systems were replaced by real-valued vectors. In the following sections, the term 'XCS' was used instead of the more appropriated term 'XCS-R' introduced in [**?**].

3.1. ICU

3.1.1 Overview of the algorithm

To discover a classification rule, ICU uses a Michigan-like learning classifier system representation, in which each rule is encoded by one individual [**?**]. Moreover, separate pools are used for each class to discover classification rules. Only one rule is kept for each class at the end of a run. Steady-state strategy is used for reproduction and replacement. The pool contains only 100 to 200 individuals for the discovery of each class and 2000 generations were used. The most common values for the crossover and mutation rate were tested (e.g. $(\chi; m) = (0,8; 0,05)$). After having tested a number of selection methods as randomly, elitism, roulette wheel and tournament, a roulette wheel selection based on the rank in the pool of the individual's fitness was retained for the replication of individuals and tournament for generational replacement.

Classification rules are symbolic expressions and describe conditions to be held and actions to be taken if the conditions are satisfied. From a functional point of view, a rule represents a piece of knowledge about a class by a conditional expression, such as *if* <*conditions*> *then* <*class*>. The "*condition*" part of a rule specifies a constraint in the system such as value, color, form, shape, etc, corresponding to conditions that must be fulfilled in order to activate the rule. The "*class*" part defines the class of the instance currently treated by the rule given the appropriate conditions are satisfied. One asserts that the evolved rules should be rapidly evaluated and easy to interpret by any user. As a result, condition representation using the concept of an interval could be fully adequate for remote sensing image classification. In terms of machine learning, the rules have to be absolutely specific, meaning that they have to cover the extreme maximum and minimum pixels belonging to any given class. A short discussion about the representation of the rules follows, but we refer the reader to [**?**] for more information.

Before rule specification, recall that a pixel is encoded as a spectral vector, describing values of reflectance for the n bands of the remote sensing image, i.e. a pixel can be considered as a point in an R^n space. In our system, the condition of any rule is built on the concept of spectral intervals defining a given band corresponding to a given class. Such intervals are a pair of integer numbers. To precisely specify the class definition, a set of intervals is defined for each band of the remote sensing image. Taking into consideration all bands, the condition part is defined as a set of hyper-rectangles in a R^n space : <*condition*> $\vdash$ $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} (m_i^j \leq b_i \leq M_i^j)$ where m_i^j and M_i^j denote, respectively, the minimal and maximum reflectance values allowed for a pixel belonging to a class C for band i. Parameter k is fixed. It defines the maximum number of disjunctions allowed.

The chosen representation is mainly due to simplicity, compactness and uniform encoding of spectral constraints. During experimentation, the representation has demonstrated rapid execution of genetic operators and efficient computing.

3.1.2 Genetic operators

Rule initialization

Generally, in remote sensing, the initial population of classification rules is randomly created from raw images and given classes, and then evolved by a GA. In this study, in order to efficiently develop the classification rules, a GA initializes interval values according to spectral limits of the classes designated by an expert for valid zones of the remote sensing image. More about initialization algorithms can be found in [**?**]. Rule initialization considers the values of reflectance most frequently recorded in the spectra of a given class. Fig 1 presents spectrograms for 5 classes (shown from left to right). Like a pencil-drawing drawn several times, the figure shows the most frequent values for a given wavelength in darker tones.

Figure 1. Spectrogram representation of the data.

Initial classification rules are created based on the maximum and minimum of the most frequent values observed on the spectrogram for each class. Two algorithms for the generation of a pool of rules have been proposed, according to the expected solution given by the expert, but allowing also some diversity. The first, called *MinMax*, creates maximum intervals covering all the pixels belonging to a given class, and the second algorithm, called *Spectro*, integrates the spectral distribution density and interval partitioning. The proposed rule initialization seems to be well suited for large volumes of data considerably reducing the search space by generating initial rules close to the final solution.

Crossover operator

Crossover exchanges hyper-rectangles at a randomly selected *level* in these rules. A *level* is the depth of conjunction or disjunction in a rule, as if it was represented by a hierarchical tree, as in genetic programming. The two new rules inherit some rule conditions from each parent rules. Each result of the crossover process has to be validated. Validation of the various rule attributes (border limits violation, overpassing, etc) is carried out by a process of interval merging. However, merging not only decreases the number of intervals in the rules, but also generates some information loss. In fact, in order to avoid premature convergence of rules, it is generally important to preserve two distinct intervals instead of a single aggregated one for the following generation.

Mutation operator

The mutation operator plays a dual role in the system: (1) it provides and maintains diversity in the population, and (2) it results in a local search mechanism in conjunction with selection. Mutation randomly selects one operation in a *library of operators* and applies it to a rule selected with a given selection scheme. The selected operator can be: (1) suppression of a spectral channel (band mutation), (2) addition or suppression of a constraint, and (3) shift or cut of a constraint in the spectral definition space of the data (interval mutation). The complexity of this operator including all the sub-operators ensures that a more diverse offspring population will be obtained.

Stop criterion

The evolution process converges according to some statistical criteria indicating if the current rule is near to a global optimum or if the population of rules will not evolve anymore. In our system, not only the evolution of quality of the best discovered rule is taken into consideration, but also a minimum acceptable quality defined by a user, a process stability measure, and a maximal number of generations. If Q_0 is the fitness of the best rule in the current generation and Q_k is the quality of the best rule obtained during the last k generation, then Q_0 is compared to the mean of Q_k for P generations. The GA is stopped when the difference falls below a given threshold E.

3.1.3 Quality Measures

The measures strongly depend on the application domain. The evaluation criteria in data mining are now considered fairly standard, but to make our approach more clear the principal quality measures are detailed. In image classification, the evaluation is usually based on the confusion matrix containing the classification produced by the classifier (I_{class}) and the classification given by an expert (I_{expert}). Table 2 defines the variables necessary to compute this measure. $P_{C_i}^{C_j}$ is the number of pixels of the class C_i classified in the class C_j by the classifier. Other values are explained later.

A very popular measure is a classifier accuracy that measures the proportion of correctly classified pixels with respect to all pixels for n classes;

$$Q_{accur} = \frac{\sum_{i=1}^{n} P_{C_i}^{C_i}}{\sum_{i=1}^{n} \Omega(C_i)} \quad (1)$$

The weakness of this measure is that it only takes cor-

Expert classification (I_{expert})

Image classified by the classifier (I_rule)	C_1	C_2	$\cdots$	C_n	Q_{ppa}	# of pix. (k_i)
C_1	$P^{C_1}_{C_1}$	$P^{C_2}_{C_1}$	$\cdots$	$P^{C_n}_{C_1}$	Q^1_{ppa}	$\Omega(C_1)$
C_2	$P^{C_1}_{C_2}$	$P^{C_2}_{C_2}$			$\vdots$	$\vdots$
$\vdots$	$\vdots$		$\ddots$		$\vdots$	$\vdots$
C_n	$P^{C_1}_{C_n}$			$P^{C_n}_{C_n}$	Q^n_{ppa}	$\Omega(C_n)$
Q_{sens}	Q^1_{sens}	$\cdots$	$\cdots$	Q^n_{sens}		

Figure 2. Confusion matrix for computing the quality measures.

rectly classified pixels into consideration. To make the classification results more specific, three other related measures are defined: Q_{ppa} positive predictive accuracy, Q_{sens} sensitivity and Q_{spe} specificity.

The positive predictive accuracy, Q_{ppa}, measures the classifier reliability of correctly classifying pixels compared to all pixels associated by a classifier to a given class:

$$Q^i_{ppa} = \frac{P^{C_i}_{C_i}}{\sum_{k=1}^{n} P^{C_k}_{C_i}} \quad Q^i_{sens} = \frac{P^{C_i}_{C_i}}{\sum_{k=1}^{n} P^{C_i}_{C_k}} \quad Q^i_{spe} = \frac{\sum_{j=1,j\neq i}^{n}\sum_{k=1,k\neq i}^{n} P^{C_j}_{C_k}}{\sum_{j=1,j\neq i}^{n}\sum_{k=1}^{n} P^{C_j}_{C_k}}$$

The sensitivity (Q_{sens}) measures the fraction of correctly classified pixels with respect to all pixels classified by an expert to a given class (i.e. proportion of true positives). The specificity (Q_{spe}) measures the rate of correctly classified pixels not being in the given class (i.e. proportion of true negatives).

The four above defined measures may give a faithless image of classifier performance in case of non equal distribution of pixels over a set of classes. Therefore, it is recommended to use an additional measure of weighted accuracy computed as follows:

$$N^i_{final} = \alpha Q^i_{sens} + (1-\alpha) Q^i_{spe} \tag{2}$$

Parameter α allows the adjustment of the relative weight given to true positives and true negatives. As mentioned before, the measure is used for certain classes that are under- (stronger α) or over-represented (lower α). By default the value of this parameter equals to 0.5, which means that the same importance is given to both measures. The proposed measure has a number of advantages; mainly it is independent of the pixel processing sequence, invariant of the size of classes, and effective for class discovery with a highly variable number of pixels. The global quality measure is a weighted average of the N^i_{final} by the size of each class.

$$N_{global} = \frac{\sum_{i=1}^{n} \Omega(C_i) N^i_{final}}{\sum_{i=1}^{n} \Omega(C_i)} \tag{3}$$

All these quality measures have been tested for different values of α and were applied on confusion matrices.

3.1.4 Classifier selection

A pixel passed to the system may activate several classifiers. The classifiers are evolved independently but during the classification process a pixel may activate several rules. At the end of a process, a selection mechanism is needed to choose the appropriate rule when more than one rule is available for a given pixel. Two methods have been developed that can be used also if no rules are activated by the pixel. Note that this operator acts only in the case when 0 or more than one rule is activated. If only one rule is activated, no modification of the result is done and the rule is kept as is.

The first one, called *BestScore*, simply selects the rule depending on the fitness value obtained from the last pass in the learning set. Among the n activated rules the rule with the best fitness is selected. If $n = 0$, the number of constraints n_c of the rules (i.e. for each spectral channel) accepted by the pixel is tested, and the rule with the highest n_c wins. In case of equality, the first rule is chosen. This seems to be a naive method. The problem comes from the use of the fitness value. Since the value is not directly connected to the data, a skew may be introduced in the evaluation method. In fact, the method has shown good results with expertise data coming from unsupervised learning. In this case, the algorithm should deal with very accurate classes but also noisy classes (*waste* classes). In this case, selection by the means of fitness is very useful.

To avoid the use of values dependent only on the paradigm, a second method has been designed, called *CloseCenter*. The deviation of the spectrum of the pixel compared to the mathematical center of each constraint in the rule is computed. For $n > 1$, the smallest deviation selects the gaining rule and for $n = 0$, the same technique as before is applied. Note that *BestScore* and *CloseCenter* adapt dynamically to the data (for each pixel), but as fitness and spectrum represent a continuum between two close pixels, the obtained classification is in most cases coherent.

3.2. XCS

XCS evolves a set of rules, the so-called *population of classifiers*. Rules are evolved by the means of a GA. A classifier usually consists of a condition and an action part. The condition part specifies when the classifier is applicable and the action part specifies which action, or classification, to execute. In contrast to the original LCSs, the fitness in the XCS classifier system, introduced by Wilson [**?**, **?**], is

based on the *accuracy* of reward predictions rather than on the reward predictions themselves. The reward is based on the basic fact than the class predicted by the whole pool of classifiers match the expert decision. Thus, XCS is meant to evolve not only a representation of an optimal behavioral strategy, or classification, but rather to evolve a representation of a complete payoff map of the problem. That is, XCS is designed to evolve a representation of the expected payoff in each possible situation-action combination.

Since our system is confronted with real-valued data in this study, the real-valued extension of XCS (XCS-R) introduced in [**?**] was applied. Recently, several studies were reported that show that XCS performs comparably well to several other typical classification algorithms in many standard datamining problems [**?**]. This section provides a short introduction to the XCS classifier system. For a more detailed introduction to XCS and XCS-R the interested reader is referred to the original paper on the real-valued XCS extension [**?**].

3.2.1 Overview of the algorithm

As mentioned, XCS evolves a population [P] of rules, or classifiers. Each classifier in XCS consists of five main components: The *condition part* specifies the subspace of the input space in which the classifier is applicable, or *matches*. In our real valued problem, a condition specifies a conjunction of intervals, one for each attribute. If the current problem instance lies within all specified intervals, the classifier matches. The *action part* specifies the advocated action, or classification. The *payoff prediction* estimates the average pay-off encountered after executing action A in the situations in which the condition part matches. The *prediction error* estimates the average deviation, or error, of the payoff prediction. The *fitness* reflects the scaled average relative accuracy of the classifier with respect to other overlapping classifiers.

Learning usually starts with an empty population. Given current input, the set of all classifiers in [P] whose conditions match the input is called the match set [M]. If some action is not represented in [M], a covering mechanism is applied. Covering creates classifiers that match the current input and specify the not covered actions. Given a match set, XCS can estimate the payoff for each possible action forming a prediction array P(A). Essentially, each entry P(a) reflects the fitness-weighted average of all reward prediction estimates of the classifiers in [M] that advocate classification a. The payoff predictions determine the appropriate classification. During learning, XCS chooses actions randomly. During testing, the action a_{max} with the highest value $P(a_{max})$ is chosen.

3.2.2 Rule selection

As in ICU, the same pixel may activate several rules coding for different classes. This behavior can be constrained for solving problems as *one-to-one classification*. Two methods (*MaxConfident* and *ScoredConfident*) were tested, asking the pool to give a unique class for each pixel: In the first one, only the rules which have a correct self-confidence are considered, in other terms, payoff prediction should be greater than a given threshold (fixed in our experiments to the half of the maximum value of payoff prediction for the current problem). Then the most frequent class obtained from rules which had matched the pixel is returned. In the second one, all the rules which have matched the pixel are considered. For a class c, a score is computed as follows for each rule r:

$$S_r = \frac{\sum(P_r F_r)}{\sum F_r} \tag{4}$$

where P_r is the prediction payoff of the rule r and F_r is its fitness. The action of the rule with the best S_r is returned. These two methods were experimented for different subsets of CASI data and for a pre-treated expertise set as follows: if more than one class was affected to the same pixel, only the dominant class was retained according to the percentage of its concentration given by the expert.

4. Case studies

In this paper, a remote sensing image of San Felice (Lagoon of Venice) has been chosen. This image contains multispectral data (CASI 15 bands), with 142x99 pixels, 16 bits per pixel and 1.3m terrain resolution [**?**]. The case studies consider a typical problem of classification for rural zones, with only five requested classes but including a high percentage of mixed pixels. Learning was carried out on 50% of the 1540 points. Validation was performed on the whole data. The table 3 summarize our experiments.

Comments. Results are detailed by classes because some are really difficult to deal with (in particular spaM), and decrease the global quality. It should be noticed than the 'P' at the end of a name of a class denotes a pure class (easier to learn) and a 'M' denote a mixed class (a mix between a dominant class at 50%, and several other classes, what we call a *mixel*). Classification is quite globally correct (on the confusion matrix not shown there, accuracy=81.1% and Kappa-index=0.81). The classification with XCS shows better results, but ICU classify at a comparable level, especially that XCS has spent many classifiers to separate spaP and spaM correctly.

	ICU		XCS	
Class	Q_{sens}	Q_{ppa}	Q_{sens}	Q_{ppa}
sarP	0.79	0.98	0.95	0.95
lisr	1	0.65	0.89	0.91
spaP	0.99	0.69	0.92	0.81
spaM	0.13	1	0.48	0.78
limM	0.93	0.93	0.98	0.91
TOTAL	**0.77**	**0.85**	**0.84**	**0.87**

Figure 3. Qualities measures.

5. Conclusion

Until now, applications of evolutionary methods in the domain of remote sensing image analysis are still very rare. A few recent reports have shown interesting results using genetic programming approaches [**?**]. In this paper two evolution-based classifiers have been described and compared on real remote sensing data. The case studies have demonstrated a comparable performance for the interpretation and the classification of heterogeneous and complex images (e.g. high number of mixels and noisy data), but more sudies are surely essential to have a better understanding of LCS applied to remote sensing data. Taking into consideration image complexity and the level of noisy information, the results of the applied evolutionary methods in our experiments are very encouraging. The learning time was relatively short (using a Pentium 2GHz, it lasts 5 minutes for ICU, 15 minutes for XCS). ICU discovers an "*if... then*" classification rule for each class using a fitness function based on image classification quality. In general, these rules are robust and can manage several disjunctions of constraints. They are able to describe the complexity of the analyzed spectra. XCS learns a pool of classification rules for all the classes. These rules have improved the accuracy of the expert and have been sufficiently generic for applying them on other parts of satellite images.

The representation of the rules allows the modeling of constraints adapted to the granularity of spectral reflectance. Hence, these rules-based approaches are of great interest when compared to traditional methods of classification. But further investigation of ICU and XCS learning efficiency are necessary. For instance, research about post-processing the rule base represents a great challenge for XCS [**?**]. It should also be noted that more powerful representation of the rules in ICU including spatial knowledge and constraints adapted to temporal series should give better results and yield more relevant rules.

References

[1] Discovery of classification rules: evolutionnary classifiers, software available at http://lsiit.u-strasbg.fr/afd/logiciels/icu/.

[2] Tidal Inlets Dynamics and Environment, Research Project Supported by the European Commission under the Fifth Framework Programme, contract n° EVK3-CT-2001-00064, http://www.istitutoveneto.it/tide, 2001-2005.

[3] M. J. Barnsley, P. Hobson, Z. Hesley, K. Evans-Jones, T. Quaife, P. Lewis, M. Disney, J.-P. Muller, A. Strahler, W. Lucht, and A. Hyman. Determination and validation of land-surface biophysical properties using SPOT-4 vegetation and HRVIR: Interim report. University of Wales Swansea, 1998.

[4] J. A. Benediktsson, P. H. Swain, and O. K. Erase. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 28:540–551, 1990.

[5] E. Bernado-Mansilla and J. M. Garrell-Guiu. Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. *Evol. Comput.*, 11(3):209–238, 2003.

[6] A. Bonarini. An introduction to learning fuzzy classifier systems. In *Learning Classifier Systems, From Foundations to Applications*, pages 83–106, London, 2000. Springer-Verlag.

[7] M. V. Butz, K. Sastry, and D. Goldberg. Tournament selection in XCS. *GECCO-2003: Proceedings of the Fifth Genetic and Evolutionary Computation Conference*.

[8] J. A. Gualtieri and R. F. Cromp. Support vector machines for hyperspectral remote sensing classification. *Proceedings of the 27th AIPR Workshop: Advances in Computer Assisted Recognition*, pages 221–232, October 1998.

[9] J. Horn, D. E. Goldberg, and K. Deb. Implicit Niching in a Learning Classifier System: Nature's Way. *Evolutionary Computation*, 2(1):37–66, 1994.

[10] J. Korczak and A. Quirin. Evolutionary mining for image classification rules. In *EA 2003 : 6th International Conference on Artificial Evolution*, Marseille, 2003.

[11] P. L. Lanzi. A study of the generalization capabilities of XCS. In T. Bäck, editor, *Proc. of the Seventh Int. Conf. on Genetic Algorithms*, pages 418–425, San Francisco, 1997. Morgan Kaufmann.

[12] D. LeRoux and M. Littman. Reinforcement learning using lcs in continuous state space. *Proceedings IWLCS-2004*, 2004.

[13] A. Quirin, J. Korczak, M. V. Butz, and D. E. Goldberg. Learning classifier systems for hyperspectral image processing. *Research Report, ULP-LSIIT-RR-2004-01*, 2004.

[14] B. J. Ross, A. G. Gualtieri, F. Fueten, and P. Budkewitsch. Hyperspectral image analysis using genetic programming. *Appl. Soft Comput.*, 5(2):147–156, 2005.

[15] S. W. Wilson. Generalization in the XCS classifier system. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 665–674, University of Wisconsin, Madison, Wisconsin, 22-25 1998. Morgan Kaufmann.

[16] S. W. Wilson. Get real! XCS with continuous-valued inputs. *Lecture Notes in Computer Science*, 1813:209–222, 2000.