Binary Rule Encoding Schemes: A Study Using The Compact Classifier System

Xavier Llorà
Illinois Genetic Algorithms Lab
University of Illinois at Urbana-Champaign
104 S. Mathews Ave., Urbana, IL 61801, USA
xllora@illigal.ge.uiuc.edu

Kumara Sastry
Illinois Genetic Algorithms Lab
University of Illinois at Urbana-Champaign
104 S. Mathews Ave., Urbana, IL 61801, USA
kumara@illigal.ge.uiuc.edu

David E. Goldberg
Illinois Genetic Algorithms Lab
University of Illinois at Urbana-Champaign
104 S. Mathews Ave., Urbana, IL 61801, USA
deg@illigal.ge.uiuc.edu

ABSTRACT

Several binary rule encoding schemes have been proposed for Pittsburgh-style classifier systems. This paper focus on the analysis of how rule encoding may bias the scalability of learning maximally general and accurate rules by classifier systems. The theoretical analysis of maximally general and accurate rules using two different binary rule encoding schemes showed some theoretical results with clear implications to the scalability of any genetic-based machine learning system that uses the studied encoding schemes. Such results are clearly relevant since one of the binary representations studied is widely used on Pittsburgh-style classifier systems, and shows an exponential shrink of the useful rules available as the problem size increases.

Categories & Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning–Concept Learning.

General Terms

Algorithms, Design, Theory.

Keywords

Learning Classifier Systems, Maximally General Classifiers, Compact Classifier System, Binary Rule Encoding.

1. INTRODUCTION

The work of Wilson in 1995 [6] was the starter of a major shift on the way that fitness was computed on classifier systems of the so call Michigan approach. Accuracy became a central element in the process of computing the fitness of rules (or classifiers). With the inception of XCS, the evolved rules targeted became the ones that were maximally maximally general (cover a large number of examples) and accurate (good classification accuracy) . The work presented here analyzes two different binary rule encoding schemes. Surprisingly, one of the most commonly used rule encoding scheme [2] of Pittsburgh-style systems inherently posses a bias that challenge the scalability of any system that uses such encoding. In this representation, theory show how the area of meaningful rules shrinks exponentially, leading the learning mechanism into a nail-in-a-haystack situation. Such situation can be corrected using alternating binary encoding schemes, as we show with a simple alternative binary encoding mechanism using the representation proposed by Butz, Pelikan, Llorà, & Goldberg [1].

GECCO'05, June 25–29, 2005, Washington, DC, USA.

2. BINARY RULE ENCODING

Regardless of the Pittsburgh or Michigan approach taken, a wide variety of knowledge representations have been used in the genetics-based machine learning community [4, 3, 2]. This paper focuses on the rule representation proposed by De Jong & Spears [2] widely adpted in the early works on Pittsburgh-style classifier systems. The main property of such representation is it simple mapping on binary string, when compared to the χ-ary mapping required by the initial Michigan one proposed by Holland [4] and later mainly followed by Goldberg [3] and Wilson [6]. We also analyzed the encoding scheme proposed by Butz, Pelikan, Llorà, & Goldberg [1].

The rule representation proposed by De Jong [2] is based on a finite set of attributes with a finite number of possible values, and a close world assumption. One of the main differences between this representation and the usual one used in the Michigan approach is that it holds internal disjunctions among attribute values [2]. It also presents the existence of unmatchable rules. In other words, rules that have conditions that will never be satisfied [5]. Moreover, the initial proposal by De Jong [2] assumed that rules match positive examples of the concept to be learnt. Any example not matched by a given rule set is, therefore, a negative example of such concept—or close world assumption.

3. MAXIMALLY GENERAL AND ACCURATE RULES

In order to promote maximally general and maximally accurate rules a la XCS [6], we need to compute the *accuracy* of a rule (α) and its *error* (ε). In a Pittsburgh-style classifier, the *accuracy* may be computed as the proportion of overall examples correctly classified, whereas the *error* is the proportion of incorrect classifications issued by the activation of the rule. For computation simplicity we assume $\varepsilon(r) = 1$ when all the predictions were accurate, and $\varepsilon(r) = 0$ when all were incorrectly issued. Let n_{t+} be the number of positive examples correctly classified, n_{t-} the number of negative examples correctly classified, n_m the number of times that a rule has been matched, and n_t the number of examples available. Using this values the *accuracy* and *error* of a rule r can be computed as:

$$\alpha(r) = \frac{n_{t+}(r) + n_{t-}(r)}{n_t} \quad ; \quad \varepsilon(r) = \frac{n_{t+}}{n_m} \tag{1}$$

It is worth to note that the error (equation 1) only take into account the number of correct positive examples classified[1]. This is a byproduct of the close world assumption of this knowledge representation. Once the *accuracy* and *error* of a rule are known, the fitness can be computed as follows.

$$f(r) = \alpha(r) \cdot \varepsilon(r)^\gamma \tag{2}$$

where $\gamma = 1$.

The total number of possible binary-encoded rules Σ for De Jong & Spears [2] given a given length ℓ is,

$$\Sigma(\ell) = 2^\ell \tag{3}$$

A rule is matchable if it guarantees that for each binary attribute, the two coding bits are not both `0` simultaneously. Thus, for any given binary attribute, four possible combinations are possible (`00`, `01`, `10`, and `11`), and one (`00`) needs to be avoided to guarantee that the attribute is matchable. Since the total number of attributes of the rule is $\ell/2$, the number of matchable rules $\Psi(\ell)$—the ones that none of the attributes contain the `00` combination—is:

$$\Psi(\ell) = 3^{\frac{\ell}{2}} \tag{4}$$

Hence, the size of the plateau of unmatchable rules $\Phi(\ell)$, is computed as

$$\Phi(\ell) = \Sigma(\ell) - \Psi(\ell) = 2^\ell - 3^{\frac{\ell}{2}} \tag{5}$$

After the normalization, the grow of the plateau is obvious. Such growth needs to be compared to the growth of matchable rules. The ratio between unmatchable and matchable rules $\rho(\ell)$ shows how scallable such rule encoding is. The $\rho(\ell)$ ratio may be computed using equations 4 and 5 as

$$\rho(\ell) = \frac{\Phi(\ell)}{\Psi(\ell)} = \frac{2^\ell}{3^{\frac{\ell}{2}}} - 1 \tag{6}$$

Equation 7 computes the exact ratio among unmatchable and matchable rules. Since we are interest on how $\rho(\ell)$ grows, such ratio may be approximated as follows ,

$$\rho(\ell) \approx e^{c\ell} \quad ; \quad c = \ln\left(\frac{2}{\sqrt{3}}\right) = 0.143 \tag{7}$$

Figure 1 compares the growth $\rho(\ell)$ of unmatchable rule in DeJong & spears representation to the growth of randomly guessing (the hazard equivalent) rules in the Butz, Pelikan, Llorà, & Goldberg [1] representation.

[1] We also assume that if a rule is never matched, no error is made and, hence, $\varepsilon(r) = 1$.

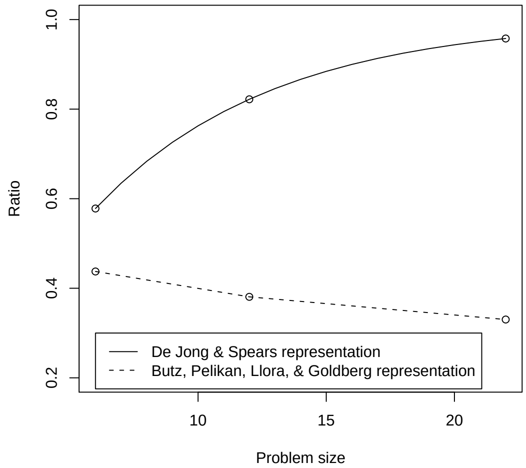

Figure 1: Figure compares $\rho(\ell)$ between the two rule encodings studied as a function of the problem size.

4. CONCLUSIONS

The binary rule-encoding representation proposed by De Jong & Spears [2] inherently posses a bias that challenges the scalability of any system that uses such encoding. In this representation, theory shows how the area of meaningful rules—the ρ ratio—shrinks exponentially, leading the learning mechanism into a nail-in-a-haystack situation. However, an alternative representation—proposed by Butz, Pelikan, Llorà, & Golberg [1]—show that such an exponentially trend is only linked to the encoding scheme used.

5. ACKNOWLEDGMENTS

This work was sponsored by USAF (F49620-03-1-0129), and by the TRECC (N00014-01-1-0175).

6. REFERENCES

[1] M. Butz, M. Pelikan, X. Llorà, and David E. Goldberg. Automated Global Structure Extraction For Effective Local Building Block Processing in XCS. IlliGAL Report No. 2005011, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2005.

[2] Kenneth A. De Jong and William M. Spears. Learning Concept Classification Rules using Genetic Algorithms. In *Proceedings of the Twelfth International Conference on Artificial Intelligence IJCAI-91*, volume 2, pages 651–656. Morgan Kaufmann, 1991.

[3] D. E. Goldberg. Computer-aided gas pipeline operation using genetic algorithms and rule learning. *Dissertation Abstracts International*, 44(10):3174B, 1983. Doctoral dissertation, University of Michigan.

[4] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[5] X. Llorà, Kumara Sastry, and David E. Goldberg. The Compact Classifier System: Motivation, Analysis, and First Results. In *Proceedings of the Genetic and Evolutinary Computation Conference (GECCO 2005)*, page in press. ACM press, 2005.

[6] Stewart W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.