Fast Rule Representation for Continuous Attributes in Genetics-Based Machine Learning

Jaume Bacardit
ASAP research group, School of Computer Science, Jubilee Campus, Nottingham, NG8 1BB
Multidisciplinary Centre for Integrative Biology, School of Biosciences, Sutton Bonington, LE12 5RD, University of Nottingham, UK
Jaume.Bacardit@Nottingham.ac.uk

Natalio Krasnogor
ASAP research group, School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK
Natalio.Krasnogor@Nottingham.ac.uk

ABSTRACT

Genetic-Based Machine Learning Systems (GBML) are comparable in accuracy with other learning methods. However, efficiency is a significant drawback. This paper presents a new representation for continuous attributes motivated by our previous work in large-scale Bioinformatics datasets, where we can observe that, very often, a very small fraction of the attributes of a domain are expressed at the same time in a rule. Automatically discovering these few key attributes and only keeping track of them contributes to a substantial speed up by avoiding useless match operations with irrelevant attributes, while potentially leading to a better learning process. The representation we propose has been tested within the BioHEL GBML system, and our experiments show that this representation has competent learning performance and reduces considerably the system run-time, up to 2-3 times faster than the state-of-the-art in fast GBML representations for datasets with hundreds of attributes.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Concept Learning, Induction*

General Terms

Algorithms, Experimentation, Performance

Keywords

Genetics-Based Machine Learning, Fast Rule Representation, Large Datasets

1 The attribute list knowledge representation

Our rule representation instead of coding all the domain attributes only keeps a list of the expressed ones, and we add (specialize) or remove (generalize) attributes from this list with a given probability. In this way, match operations only evaluate a subset of attributes (ignoring all the non-expressed attributes), possibly avoiding hundreds of irrelevant computations. Moreover, as the representation only holds relevant attributes the exploration operators will always recombine/mutate data that matters, potentially leading to a better learning process too.

Figure 1: Example of a rule in the attribute list knowledge representation with four expressed attributes: 1, 3, 4, and 7. l_n = lower bound of attribute n, u_n= upper bound of attribute n, $c1$=Class 1 of the domain

#Expr. Atts.	4							
Expr. Atts.	1	3	4	7				
Intervals	L_1	U_1	L_3	U_3	L_4	U_4	L_7	U_7
Class	C_1							

Each rule will be represented by four elements, as shown in figure 1: (1) an integer containing the number of expressed attributes, (2) a vector specifying which attributes are expressed, (3) a vector specifying, for each expressed attribute, the lower and upper bound of its associated interval and (4) the class associated to the rule. Thus, semantically a rule specifies an hyperrectangle in the search space.

In initialization, a parameter specifies the expected value of number of expressed attributes, following [3]. A probability of expressing an attribute is derived from it and a subset of attributes is added to the list of expressed ones given this probability. An interval ranging from 25% to 75% of the domain width is initialized for each expresseed attributes. Intervals are seeded from sampled training examples. The crossover operator acts as a one-point crossover, but taking into account that different parents may have different lists of expressed attributes, making sure to maintain semantical correctness. Mutation acts as in a standard GBML system. Two operators are added to the GA cycle after mutation to add attributes to the list of expressed attribute (specialize operator) or remove attributes from the list (generalize operator) to the individuals of the offspring population. In case of applying the specialize operator, a randomly initialized interval is generated for the attribute chosen to be expressed. An individual-wise probability is used to decide the application of these two operators across the population.

2 Experiments on small datasets

The first step to experimentally validate this representation is to determine whether this representation is able to learn properly when compared to state-of-the-art GBML methods. To do so, we have compared our representation against

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

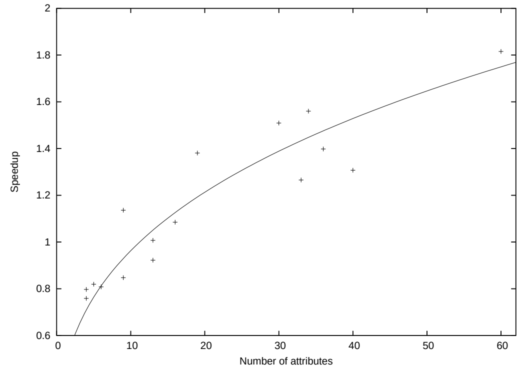
Table 1: Results of the experiments on the bioinformatics datasets

Dataset	Result	NAX	Prob. of Generalize and Specialize in Att. List KR				
			0.05	0.10	0.10	0.20	0.25
SS	Acc.	72.4±1.0	73.3±0.8	73.4±0.9	73.3±0.8	73.3±0.8	73.2±0.7
	#rules	268.7±13.6	290.9±10.4	281.6±10.3	271.4±10.3	263.4±7.8	253.3±9.1
	#exp. att.	13.1±3.0	14.6±3.2	14.4±3.2	14.1±3.2	13.7±3.2	13.4±3.2
	run-time (h)	16.1±0.9	6.4±0.4	6.0±0.6	5.9±0.6	5.7±0.4	5.6±0.4
CN	Acc.	80.9±0.4	81.1±0.4	81.1±0.4	81.1±0.4	81.0±0.4	81.0±0.4
	#rules	263.2±12.6	284.7±12.5	275.1±13.3	265.5±13.4	255.5±11.2	245.1±11.8
	#exp. att.	14.3±2.9	16.3±3.0	16.1±3.1	15.7±3.1	15.2±3.1	14.8±3.1
	run-time (h)	45.7±2.5	30.9±2.1	29.8±2.3	28.9±2.3	28.1±1.8	26.7±2.0

another recent efficiency-oriented representation, taken from the NAX system [3]. This representation uses vectorial SSE instructions to boost the efficiency of the match operations. Semantically both representations evolve identical types of rule. The representations are evaluated within the framework of BioHEL [1, 2], a recent GBML system. We have used a set of 16 small-size datasets with continuous attributes from the well-known UCI repository for this first stage of experiments. We also performed a sensitivity analysis of the probability of applying the generalize and specialize operators, testing five different probabilities in the 5%-25% range. These two operators have to identify the relevant attributes for a rule, so their correct functioning is crucial for the success of the representation.

The results of these experiments indicate that (1) our representation obtains similar accuracy to the NAX representation, according to a Friedman statistical tests for multiple comparisons. (2) the sensitivity analysis indicated that the different probabilities evaluated gave similar accuracy results (3) in relation to run-time, the NAX representation was faster than our representation in the datasets with smallest number of attributes, and our representation became faster and faster with increasing number of attributes. Figure 2 plots the speedup of our representation (with 25% of probability of generalize/specialize) over NAX against the number of attributes of the domain. We have fitted a small speedup model using R of the kind $speedup = a \cdot \sqrt[3]{N}$ where N is the number of attributes of the domain.

Figure 2: Speedup of our representation over NAX plotted against number of attributes



3 Experiments on large datasets

We have used two of our large-scale bioinformatics datasets to evaluate the full potential of the representation. The two datasets belong to the protein structure prediction (PSP) family of problem [4]. The first dataset, called Secondary Structure (SS) prediction has 83823 instances and 300 attributes. The second dataset, called Coordination Number (CN) prediction has 257560 instances and 180 attributes. Table 1 shows, for each of these two datasets the obtained accuracy, average rule-set size, average number of expressed attributes per rule and run-time (reported in hours)[1].

As we expected, this representation is able to explore better the search space because it only needs to recombine relevant attributes. This happens specially in the SS dataset, where the representation manages to obtain an accuracy 1% higher than the NAX representation. In the CN dataset our representation also obtains higher accuracy, although the difference is minor. Higher probabilities of generalize and specialize obtain more run-time reduction and generate more compact solutions with smaller rule sets and expressed attributes per rule. In the SS dataset we have managed to reduce the average run-time from more than 16 hours to less than 6 hours. Our representation is almost three times faster than NAX. On the CN dataset our representation is up to 1.7 times faster than NAX. The run time of our representation is up to 19 hours shorter.

4 Conclusions

Our representation is able to learn equal or better than other recent GBML alternatives, and it manages to substantially reduce the training time for large datasets with hundreds of attributes. Thus, we can say that the objectives that we had for designing this representation have been fulfilled.

Acknowledgments

We would like to acknowledge Michael Stout and Jonathan D. Hirst for their help in relation to the Bioinformatics datasets and the support of the UK Engineering and Physical Sciences Research Council (EPSRC) under grant GR/T07534/01.

5 References

[1] J. Bacardit and N. Krasnogor. Biohel: Bioinformatics-oriented hierarchical evolutionary learning. Nottingham eprints, University of Nottingham, 2006.

[2] J. Bacardit, M. Stout, J. D. Hirst, K. Sastry, X. Llorà, and N. Krasnogor. Automated alphabet reduction method with evolutionary algorithms for protein structure prediction. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 346–353, New York, NY, USA, 2007. ACM Press.

[3] X. Llorà, A. Priya, and R. Bhargava. Observer-invariant histopathology using genetics-based machine learning. *Natural Computing, Special issue on Learning Classifier Systems*, page in press, 2008.

[4] M. Stout, J. Bacardit, J. D. Hirst, and N. Krasnogor. Prediction of recursive convex hull class assignments for protein residues. *Bioinformatics*, 24(7):916–923, 2008.

[1] Experiments were run on Opteron processors running at 2.2GHz, Linux operating system and the C++ implementation of BioHEL.