Published as: Madden, A.D. (1999). **Genetic algorithms: a pragmatic, non-parametric approach to exploratory analysis of questionnaires in educational research.** *Educational Research*. 41 163 – 172.

Abstract

Data from a survey to determine student attitudes to their courses are used as an example to show how genetic algorithms can be used in the analysis of questionnaire data. Genetic algorithms provide a means of generating logical rules which predict one variable in a data set by relating it to others. This paper explains the principle underlying genetic algorithms and gives a non-mathematical description of the means by which rules are generated. A commercially available computer program is used to apply genetic algorithms to the survey data. The results are discussed.

Introduction

The aim of this paper is to provide the educational researcher with a practical introduction to a versatile analytical technique. Genetic algorithms have been around for over twenty years and are finding use in a growing number of areas but not, to date, in educational research. The paper uses as an example, data collected from a survey of students being introduced to computer assisted learning courseware. These data were not specifically selected; they were used merely because they were available for use without permission being needed.

The name ‘genetic algorithm’ is initially misleading, suggesting as it does, some biological association. However, as is explained below, the name is metaphorical and the technique is applicable to data of many types and many forms.

Predictive rules

If there is reason to suppose that the variables in a data set are related in any way, it may be possible to predict or classify one variable by using some or all of the others. The predictors/classifiers can then be combined using logical and arithmetic operators to produce predictive rules (Fig 1).

A hypothetical example is of a teacher with a set of data relating to student interests. The teacher may wish to see if the data can be used for predicting whether or not a given student will opt to study science at A Level. The data set includes information about attitudes towards puzzles, maths and computer games. Examination of the data reveals that if a student is very keen on all three, there is a 79% chance that he or she will choose a science A Level.

With a simple data set, such relationships are likely to be immediately obvious. In other, more complex data sets, the relationships can be hard to spot. Statistical analyses may be used, but the results are often hard to interpret or, in the case of parametric tests, may be inappropriate. This paper looks at a means of generating and testing predictive rules and applies it to the results of a survey.

Genetic algorithms

Evolution by natural selection - the model for genetic algorithms

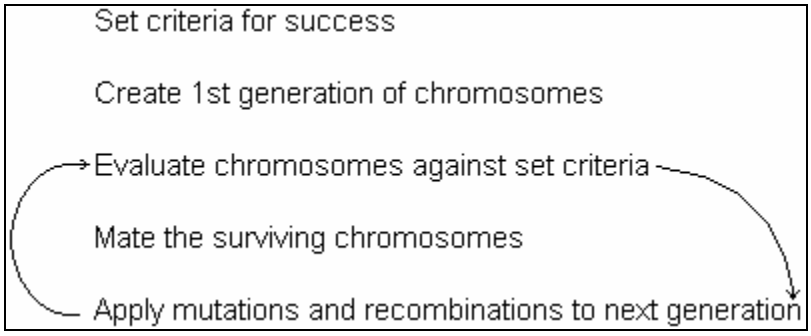
Evolution by natural selection through survival of the fittest is nature's way of solving problems. Every organism is a solution to the problem of surviving and breeding in a given habitat where resources are limited. The organisms with the best solutions to the problem (i.e. the fittest) will produce the greatest number of surviving offspring.

Solutions are passed on from one generation to another in the form of a genetic code. Successful parents will pass that code on to their progeny. However, other factors will affect the genetic code of the progeny. Where there is sexual reproduction, new combinations of genetic material will arise. In addition, mutations will inevitably occur.

The offspring will therefore differ from their parents in many respects. The characteristics which helped each parent to survive may, when combined in new ways, lead either to better or to worse solutions to the problem of survival. However, given that the parents had been the fittest of their generation, the overall fitness of each generation is likely to be greater than that of the previous generation (Fig 2). The population fitness as a whole will gradually approach an optimum (Fig 3).

Genetic algorithms as an approach to solving more general problems

In the early 1970's, John Holland had the idea of applying the evolutionary ideas described above to the solution of a wider range of problems. If the problem can be expressed as a primary event and a set of predictive rules, the logical and arithmetic operators used in the rules can be coded. Holland, in keeping with the analogy of natural selection referred to the resulting codes as chromosomes. He devised a rule set (or algorithm) for testing and varying the chromosomes. The genetic algorithm developed can be summarized as follows:



In the example of the teacher, the criterion for success is *"Will this student choose to study science?"* The rules which produce the most reliable predictions are combined (mated) at random to produce variations. These enter the rule population and the process repeats.

As with all analogies, the comparison between genetic algorithms and the evolutionary process breaks down at certain points. Evolution is a dynamic process: no niche is entirely stable. An organism which survives in a niche today may be less well suited to that niche than one of its extinct competitors might have been. At the time of the competition however, the niche being competed for was slightly different. Evolution is therefore an attempt to solve a problem which changes at random over time.

Genetic algorithms by contrast, are evolving towards a stable niche. Although they are initially generated at random, the criterion for success remains the same: i.e., that they successfully predict the primary variable of the given data set by using the named secondary variables. A successful rule therefore, will always be successful. It should be remembered however, that because the rules are randomly generated within a particular data set, they do not necessarily relate to any underlying pattern and may merely be an artefact, relevant only to the set within which they were generated.

Methodology

The data analysed came from a survey of student attitudes towards their courses. Students were asked to state their age, gender, year of study, details of their qualifications, and whether or not they had children. They were then asked for their views on a series of statements related to their courses. For the purpose of this

example, an attempt has been made to generate a profile of the students most likely to give a particular response to two of the more general statements on the survey:

1. I have acquired skills that generally will be useful to me outside the University
2. I have received enough information to enable me to make the right choices about optional units.

The part of the survey form analysed in this paper is summarized in Table 1.
The results of the survey were analysed using a DOS-based genetic algorithm program called BEAGLE (Bionic Evolutionary Algorithm Generating Logical Expressions), written by Richard Forsyth at the University of the West of England in Bristol.

BEAGLE cannot handle missing data, so after appropriate adjustment the data set comprised 604 students for statement 1 and 611 students for statement 2. Beagle offers the option of splitting the data set so that half of the data can be used for generating rules, and the remaining data can be used for testing them. This option was used, and the rules for statements 1 and 2 were generated from 297 and 301 data respectively.

In this instance, BEAGLE was used to profile the students most likely to be discontented: i.e., those who disagreed with one or other of the two statements, so the rule file consisted of

Statement > *3*.

The program generated 20 predictive rules at random (Table 2). These were then subjected to the genetic algorithm described above for 200 generations. The most successful rule was saved, and the variables used in this rule were removed from the analysis. The process was repeated until all the variables had been incorporated in

rules. BEAGLE then either discarded the resulting rules as being not significant, or wrote them to a file. In this example, no more than two rules were retained.

Interpreting the output

Rules such as 1, 2 and 8 in Table 2 are straightforward. Other rules appear to be either very complicated or nonsensical. It should be remembered that the rules have all been generated at random, so it is quite possible for them to be nonsensical. Rules such as

(Year > 0) (rule 14)

and *(-103.8491 > 0)* (rule 15)

fall into this category and illustrate one of the risks of using an inappropriate test criterion. For rule 14, the outcome is always TRUE, while for rule 15 the outcome is always FALSE. If the target had been

S1 < 5,

it would have been true in 587 cases out of 604 (97%), so rule 14 would have been a good predictor.

BEAGLE-generated rules are further complicated by the fact that BEAGLE uses logic and arithmetic interchangeably. It uses the standard conversion:

True = 1, False = 0.

Where a numerical input is required, either 1 or 0 is entered, as appropriate. Where an expression is in two or more parts therefore, the parts will be evaluated and compared, as dictated by the parentheses. For example, Rule 1 of Table 3

((age + year) < 4.0000) = children)

suggests that, if the respondant is a second year student of under 21 with pre-school children, the overall result will be False. This is derived as follows:

Pt1: *(age + year)* *= (2 + 1) is less than 4, ∴ Pt1 = 1 (= True)*

Pt2 *children* *= 2*

so Pt1 ≠ Pt2, and the overall value for the expression is 0 (= False). By working through the possible values of each of the variables in this way, it is possible to generate truth tables (eg, Tables 7 and 8) which can be used to make predictions about the data set being examined.

Statistics for the rules

The row of figures beneath each rule indicates how successful the rule is. The first value is derived from a standardized χ^2 value. The maximum possible score is 100. The next four figures are the contingency table (Table 3), which shows the numbers of correct and incorrect predictions.

Once a rule set has been generated, Beagle can be used to test the rules in combination. Any that do not make a significant contribution can be dropped. In this example, no more than two rules were generated (Tables 4 and 5), and in all cases, Beagle recommended dropping the less successful rule.

Avoiding local optima

Because rules are generated at random, there is a slight chance that the best rules will be missed. To reduce this, BEAGLE was run a further two times and the most successful of the three rules was used.

In assessing the success of a rule, the user should consider not only its predictive power, but also the number of variables used in it. Where the inclusion of further variables results in only a slight increase in the χ^2 value, it may be better to use the simpler rule.

Results

Statement 1

The most successful rule generated in each of the three runs for statement 1 is, in effect, the same in all cases. This is indicated by the accompanying statistics and can be confirmed by generating truth tables such as Tables 7 and 8.

The easiest form in which to use the rule for statement 1 is:

((qualif = 1.0000) = gender).

This will only be true for female students with A'Levels (see Table 6) suggesting that they are more prevalent amongst students who feel that they have failed to acquire skills of use to them outside the University than are other groups of students.

This prediction was tested on the test data for statement 1 generated by BEAGLE. The χ^2-test in Table 8 shows that significantly more females with A'Levels were dissatisfied than those without.

Statement 2

The three runs for statement 2 generated three different rules, of which the first, ***((year < age) <> (3.000 > year))***, was the most successful. From the truth table generated for this rule (Table 8), it is possible to predict that the following groups will not feel that they have received enough information to enable them to make the right choices about optional units:

- First and second year students under 21
- Second year students of under 30

As shown in Table 10, there is a significant effect due to year of study and age. Most of this effect is due to the large number of dissatisfied second year students who were under 21 at the time they started their course. Contrary to the prediction, made above however, there are slightly fewer dissatisfied young first years and considerably more dissatisfied young third years than might have been expected. Nevertheless, BEAGLE has been useful in pointing to the fact that 41% of second year students feel that they have not received enough information, compared to 29% of the student population as a whole.

Conclusion

The example above gives some idea of the versatility of genetic algorithms as a means of exploratory analysis. On their own, the rules generated are often nonsensical, but if the genetic algorithms are applied in association with an understanding of the data being studied, some valuable insights can be gained. Furthermore, because those insights are based on logic rather than statistics, they are likely to be easier to understand and to translate into useful predictions.

References

Davis, L. (1991): Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York

Forsyth, R.S. (1986): BEAGLE User Guide. Warm Boot Ltd.

Holland, J.H. (1975): Adaptation in Natural And Artificial Systems. University of Michagan Press.

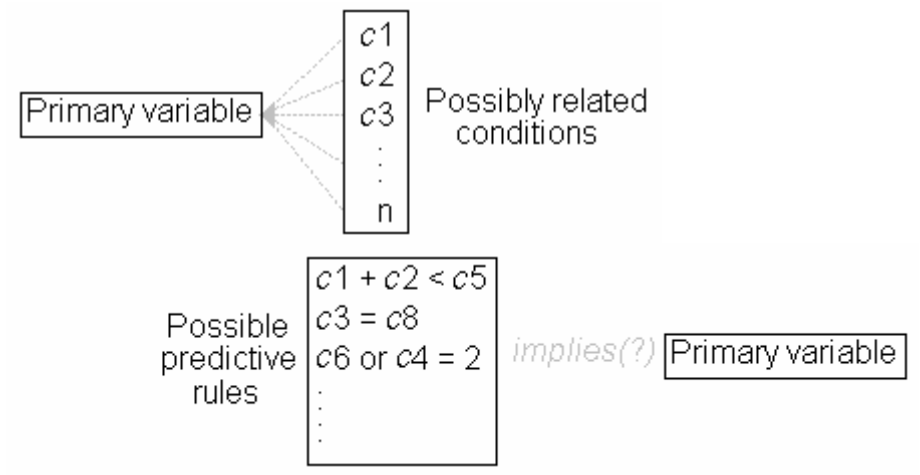


Fig 1: Using associated information to generate predictive rules

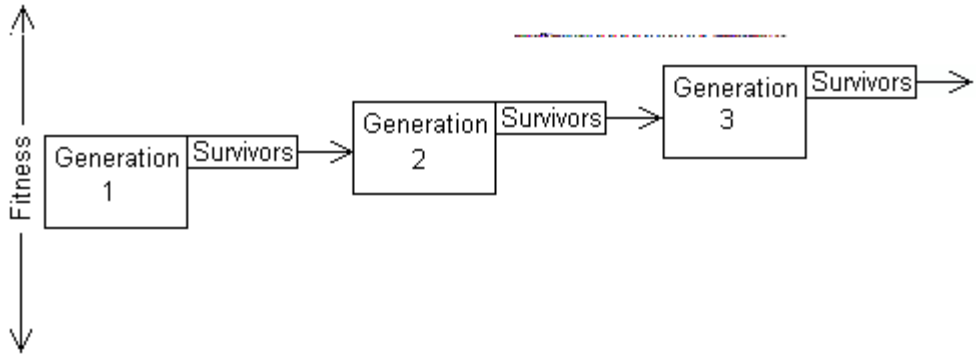


Fig 2: Increase in population fitness in successive generations

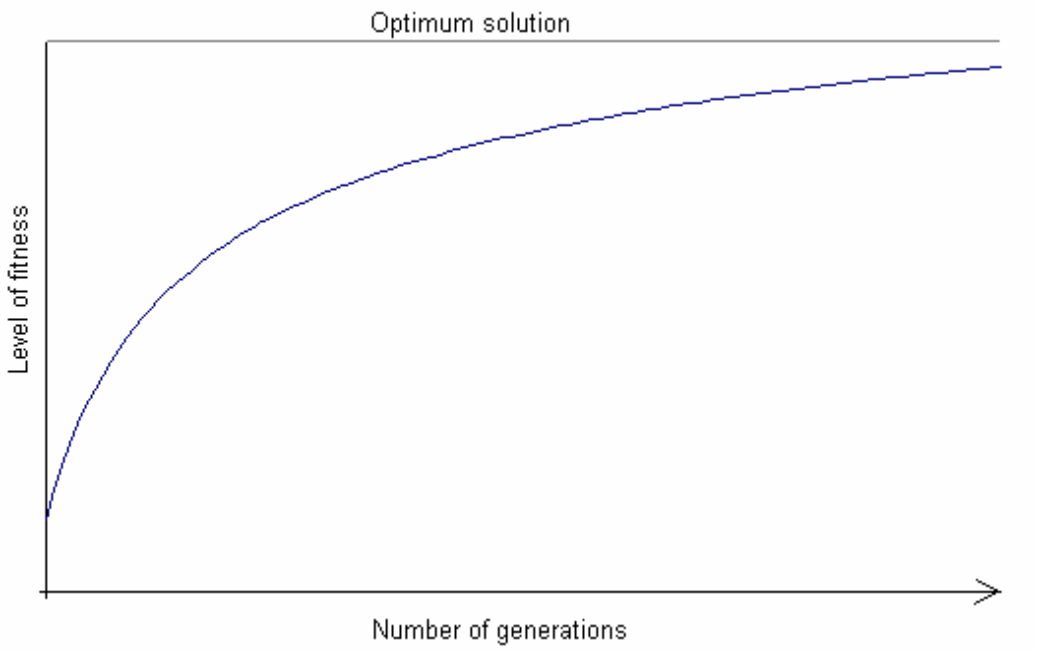


Fig 3: Population fitness approaches an optimum

Scoring	1=	2=	3=
Year of study	1st	2nd	3rd
Age at start of course	< 21	≥ 21 > 30	≥ 30
Gender	Female	Male	N/A
Age of children	No children	Pre-school	At school
Qualifications	A'Levels	O'Level/GCSE + other	Other
Q1: I have acquired skills that generally will be useful to me outside the University Q2: I have received enough information to enable me to make the right choices about optional units.			

Table 1: Summary of the information requested from students. They were asked to respond to the questions by scoring them on a scale of 1 to 5, where 1 = strongly agree, 3 = neutral, and 5 = strongly disagree.

	Rule = (S1 > 3.0000)$
1	(age < 3.0000)
	$ 7.24 38 225 2 32
2	(qualif < age)
	$ 7.04 4 48 36 209
3	((year >= (children + 1.0000))& (year > (children + 1.0000)))
	$ 6.60 1 27 39 230
4	(gender < (year - 1.0000))
	$ 6.49 1 22 39 235
5	! ((gender <= qualif)& (gender >= -88.7487))
	$ 6.41 5 58 35 199
6	((children <= gender)& (qualif <= gender))
	$ 4.36 35 207 5 50
7	!((qualif <= (children + 0.0000))& (gender <> (children - 1.0000)))
	$ 4.21 4 46 36 211
8	(children > 1.0000)
	$ 4.11 2 23 38 234
9	(qualif <= year)
	$ 2.55 36 223 4 34
10	((qualif <= 2.0000)| (year >= (age + 0.0000)))
	$ 1.87 39 244 1 13
11	(qualif >= year)
	$ 1.76 23 157 17 100
12	(qualif > (year + -1.0000))
	$ 1.26 23 157 17 100
13	((children >= -27.9539)& (gender >= year))
	$ 0.95 24 164 16 93
14	(year > 0)
	$ -0.25 40 257 0 0
15	(-103.8491 > 0)
	$ -0.25 0 0 40 257
16	(-68.7478 > 0)
	$ -0.25 0 0 40 257
17	(gender <= 78.9761)
	$ -0.75 40 257 0 0
18	(children >= (year + 81.8572))
	$ -1.25 0 0 40 257
19	(year < (gender * -100.6885))
	$ -1.25 0 0 40 257
20	((children > -112.5618)& (year > -9.8939))
	$ -1.75 40 257 0 0

Table 2: Rules randomly generated by BEAGLE

	Actual:	True	False
Predicted: (by Rule2)	True False	4 36	48 209

Table 3: Contingency table

Run	**(S1 > 3.0000)**
1	((qualif = 1.0000) = gender) 14.91 31 139 9 118 (((year + age) < 3.5000) >= children) 13.31 36 181 4 76
2	((gender < 1.5000) >= qualif) 14.91 31 139 9 118 (age <= (year <= 2.0000)) 13.24 33 160 7 97
3	(gender = (qualif <= gender)) 14.91 31 139 9 118 (((age + year) < 4.0000) = children) 13.31 36 181 4 76

Table 4: Rules evolved by BEAGLE for statement 1

Run	**(S2 > 3.0000)$**
1	((year < age) <> (2.5000 > year)) 18.57 75 122 19 85
2	((year >= 2.5819) < ((age > year) = (age <= (year >= 2.5819)))) 17.07 75 122 19 85 ((qualif <= gender) = children) 12.09 80 151 14 56
3	(age = (year <= 2.0000)) 15.39 67 111 27 96 ((children - (gender >= qualif)) <= 0.0000) 12.09 80 151 14 56

Table 5: Rules evolved by BEAGLE for statement 2

((qualification = 1) = gender)		
	M	**F**
A'Levels	F	T
O'Level/GCSE + other	F	F
Other	F	F

Table 7: Truth table for statement 1 rule

((year < age) <> (3.000 > year))			
	< 21	**21 - 30**	**≥ 30**
First	y	n	n
Second	y	y	n
Third	n	n	n

Table 8: Truth table for statement 2 rule

	Satisfied		**Dissatisfied**		**Total**
	O	**E**	**O**	**E**	
Women with A' Levels	135	139	21	17	156
Women without A'Levels	36	32	0	4	36
Total	171		21		192
χ^2 for 1 d.f. = 5.441 (significant at 5%)					

Table 9: χ^2-test of prediction made by rule for statement 1, that female students with A'Levels are more likely to feel that they have failed to acquire skills of use to them outside the University than are other groups of students. (Observed values given to nearest whole number)

	Satisfied		Dissatisfied		Total
	O	**E**	**O**	**E**	
1st years < 21	88	86	34	36	122
2nd years < 21	43	53	32	22	75
3rd years < 21	20	15	1	6	21
1st years ≥21< 30	15	15	6	6	21
2nd years ≥21< 30	10	11	6	5	16
3rd years ≥21< 30	4	4	2	2	6
1st years ≥ 30	18	15	3	6	21
2nd years ≥ 30	9	10	5	4	14
3rd years ≥ 30	12	10	2	4	14
Total	219		91		310
χ^2 for 8d.f. = 17.333 (significant at 5%)					

Table 10: χ^2-test of prediction made by rule for statement 2, that first and second year students under 21, and second year students under 30 are more likely to feel that they have not received enough information to enable them to make the right choices about optional units. (Observed values given to nearest whole number)