Genetic Programming for Association Rules on Card Sorting Data

Michelle Lyman and Gary Lewandowski
Mathematics and Computer Science
Xavier University
Cincinnati OH 45207-4441
{lyman, lewandow}@cs.xu.edu

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Experimentation

Keywords

data mining, genetic programming, card sorts

1. INTRODUCTION

Personal Construct Theory [2] describes the basic notions behind a constructivist theory of understanding and learning. The theory suggests learning is a process of building constructs that link concepts together in a way that satisfies current knowledge about the concepts and that allows them to deduce new ideas from the construct. For researchers seeking to understand how people view or understand the world, eliciting these constructs is important.

Card sorting is one technique used to elicit conceptual constructs. Used in a variety of studies ranging from Human Computer Interaction [8] to requirements engineering [3] to computer science education [7], the technique elicits conceptual structures from participants by asking them to sort a group of cards with concepts written on them. After the participants sort the cards, they are asked to give a name to each group and to the overall sorting criteria they used. Researchers study a variety of data resulting from the sorts, including the number of sorts and the number of groups within sorts, the similarity in group or criteria names across subjects, and the sense of distance of concepts from each other when examining how concepts are grouped together in the sorts.

Most card sort studies have been relatively small, allowing researchers to examine the sorts completely by hand . We are interested in helping researchers study card sort data in much larger studies, motivated by a recent study by Petre et al [7]. This study on conceptual understanding of programming involved twenty schools, 275 subjects, 1258 sorts (criteria) and approximately 5000 groups. In a dataset of this size it is easy to get basic information on the number of sorts and groups, but it is particularly difficult to get a meaningful sense of linked concepts.

1	function	10	scope	19	type
2	method	11	list	20	loop
3	procedure	12	recursion	21	expression
4	dependency	13	choice	22	tree
5	object	14	state	23	thread
6	decomposition	15	encapsulation	24	iteration
7	abstraction	16	parameter	25	array
8	if-then-else	17	variable	26	event
9	boolean	18	constant		

Figure 1: Stimuli used in card sort task.

As with other large datasets, it is useful to have a tool that helps find potentially interesting aspects of the data which can then be examined more thoroughly by researchers. In this abstract we present an overview of a genetic programming approach to the problem of analyzing card sorting data. We focus on the computer programming concepts study to illustrate the approach, but the techniques are applicable in any card sort study.

2. PROGRAMMING CONCEPTS STUDY

The programming concepts study by Petre et al. [7] is the largest known card sorting exercise to date. The study was conducted on both educators and ”first competency programmers”, defined as those who were considered to have seen enough material to be presented with problems from a study by McCracken et al [5] on beginning programmers. Typically this meant subjects had completed one or two programming courses. Subjects were given twenty-six concept cards (listed in Figure 1) and asked to sort them from the point of view of writing a program. Along with the sort information, interviewers recorded other information including a general rating of their performance in each course they had taken.

3. ASSOCIATION RULES

Association rules, introduced by Agrawal et al [1], are composed of an antecedent, a, and a consequent, c in the form, ”If a, then c,” where a and c both evaluate to Boolean values. The support for a rule is the percentage of the population fulfilling the antecedent and the consequent. The confidence for the rule is the percentage of the population fulfilling the antecedent that also fulfills the consequent. Intuitively, the support is a measure of the size of the sub-

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

population the rule applies to while the confidence is a measure of how well the antecedent predicts the consequent. A conjunctive association rule uses the Boolean AND operator to relate additional data. Mata, Alvarez, and Riquelme [4] mined conjunctive association rules using a genetic algorithm. A disjunctive association rule introduces the Boolean operators OR and XOR. Nanavati, Chitrapura, Joshi, and Krishnapuram [6] used an algorithm called Thrifty-Traverse to mine disjunctive association rules.

4. GP FOR CARD SORT RULES

The main research task in constructing a GP for mining disjunctive association rules from card sorts is allowing rules to be constructed that recognize the two levels of information in the sort, namely the cards that are placed together in groups, and the larger relationship of cards between groups in a sort. For example, it would be useful to know not only that LIST and ITERATION are frequently placed in the same group, but also that in sorts of that nature, another group typically holds RECURSION, TREE and ABSTRACTION. Thus the rule tells us both about cards typically placed in the same group and also about commonalities over an entire sort.

We introduce operations to operate at two levels of organization, appropriate for any card sorting task, and also useful in any task where one considers multiple levels of organization. Operators on groups are called "G operators" while operators on the sorts are called "S operators." We allow only AND as a G operator, but allow AND, OR, and XOR as S operators. S operators must have S or G operators as children. G operators must have G operators or cards as children. Our GP also enhances the notion of fitness in a disjunctive association rule by introducing the notion of *balance* to help us detect cases in which the evolved rule has many disjunctions that apply to only a few population members but boost the fitness level. Unbalanced disjunctive rules have high confidence and support but do not actually reveal useful information.

We use ramped-half-and-half to generate an initial population of rules. The genetic operations selection, mutation, and crossover guide the evolution of each rule. The top ten percent of the population automatically survives to the next generation. In the remaining ninety percent, selection of rules is based on the roulette wheel approach, where each rules chance of selection is based on its fitness. Once selected, rules are either mutated or crossed over; the choice to mutate versus crossover occurs with a probability selected via a user-parameter.

The fitness of a rule is the product of the confidence of the rule, the support of the rule, the percentage of the cards involved in the rule, and the balance of the rule. The balance is defined as follows. Card nodes and G-operator nodes have balance 1. S-operator node balance is calculated as follows:

1. Calculate $left$ and $right$, the number of sorts satisfying the left and right subtrees, respectively.

2. Calculate the basic balance of the node as $2 * min(left, right)/(left + right)$. This factor is 1 if an equal number of sorts satisfy each subtree.

3. Multiply the basic balance by the smaller of the balance of the left and right subtrees, thus accounting for imbalances at lower levels of the rule.

5. RESULTS

The rules discovered for the educators subpopulation reveal that more than 50 percent of the sorts by educators group in one of two distinct ways; the first is a group of lower-level, concrete concepts, the other is higher-level abstract concepts. Only the educators yielded a rule with five cards in a single group, suggesting that educators have more agreement in their grouping of the cards. Furthermore, except for the high performance students, no subtrees have more than three cards and all require at least three XORs to achieve the level of support and confidence necessary to make an acceptable rule. The concepts grouped in the high performers group are similar to the higher-level concepts grouped by educators. Examining the original data to see the names of the groups holding these concepts indicates that the students group them in opposition to data structures or other concrete notions such as operations or languages.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NSF MRI grant 0215836 and DUE grant 0122560. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank the Bootstrapping Research in Computer Science Education workshop for providing the card sorting data.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International conference on Management of data*, pages 207–216, 1993.

[2] G. Kelly. *The psychology of personal constructs.* Norton, 1955.

[3] N. M. Maiden and M. Hare. Problem domain categories in requirements engineering. *International Journal of Human-Computer Studies*, 49:281–304, 1998.

[4] J. Mata, J. Alvarez, and J. Riqueime. An evolutionary algorithm to discover numeric association rules. In *Proceedings of SAC 2002*, 2002.

[5] M. McCracken and et al. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4):125–140, December 2001.

[6] A. Nanavati, K. Chitrapura, and J. Krishnapuram. Mining generalised disjunctive association rules. In *CIKM '01*, 2001.

[7] M. Petre, S. Fincher, J. Tenenberg, R. Anderson, R. Anderson, D. Bouvier, S. Fitzgerald, A. Gutschow, S. Haller, G. Lewandowski, R. Lister, R. Mccauley, J. McTaggart, B. Morrison, L. Murphy, C. Prasad, B. Richards, K. Sanders, T. Scott, D. Shinners-Kennedy, L. Thomas, S. Westbrook, and C. Zander. 'my criterion is: Is it a boolean. a cardsort elicitation of students knowledge of programming constructs". Technical Report 6-03, Computing Laboratory, University of Kent, Canterbury, UK, 2003.

[8] L. Upchurch, G. Rugg, and B. Kitchenham. Using card sorts to elicit web page quality attributes. *IEEE Software*, pages 84–89, 2001.