Appears in Z Ras (ed), Lecture Notes in Computer Science, Vol. 0869, Springer-Verlag, 1997; also appears in the Proceedings of the 10th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October, 1997.

97-6

Speeding GA-Based Attribute Selection for Image Interpretation

Qi Zhang

Machine Learning and Inference Laboratory George Mason University Fairfax, VA 22030, USA qzhang@aic.gmu.edu

Abstract

This paper addresses the problem of GA-based attribute selection. Previous work in this direction has mainly focused on problem representation so that a genetic algorithm could work on it searching for a satisfactory attribute subset. Even though good experimental results were reported, they were usually acquired at the cost of time. This paper presents a novel approach to this problem. In particular, it introduces attribute quality measure during genetic evolution in order to make some promising attributes more likely to appear in a new generation. In this way, the evolution process is faster, and satisfactory results can be achieved in less time. Preliminary experimental results in image interpretation show that this approach is promising.

KEYWORDS: evolutionary computation, attribute selection, image interpretation.

1 Introduction

Image interpretation has important military and civil applications such as detecting military targets in images taken from airplanes or finding tumors in medical images. A set of attributes of high quality for describing objects is crucial to the recognition or interpretation performance of such systems.

Defining good attributes is problem-dependent and is beyond the scope of this paper. Of particular interest here is how to select a satisfactory subset from candidate attributes because it is known that (1) there is nonlinear interaction among attributes such that no two attribute subsets have the same discriminating power and (2) simply adding an attribute can possibly degrade system performance (e.g., Kreithen, 1993). In addition to possible better results, another benefit from attribute selection is that it can make a system work faster due to fewer attributes.

Many researchers contributed to the area of attribute selection (e.g., Forsburg, 1976; John et al., 1993; Koller and Sahami, 1996; Vafaie and De Jong, 1993; Bala et al., 1995). Recently, the GA-based approach to this problem has attracted some researchers (e.g., Vafaie and DeJong, 1993; Bala et al., 1995). This approach seems promising and good results were reported (e.g., Bala et al., 1995). However, these results were usually acquired at the expense of time spent in genetic evolution. In this paper, we propose a novel way of using genetic algorithms to address attribute selection, which achieves good results without consuming much time. We introduce easily available attribute quality measure into GA evolution to speed the process of

searching for a satisfactory subset from candidate attributes. This idea is demonstrated in a multistrategy learning system which combines a genetic algorithm (De Jong, 1996) and the inductive learning program AQ15c (Michalski et al., 1986, Wnek et al., 1995). Specifically, it starts with a statistical analysis of candidate attributes and selects some potentially promising ones and makes them more likely to appear in the first generation of genetic evolution. As for an evolved attribute subset, AQ15c is called to perform learning and testing upon actual image data. The testing accuracy is taken as the fitness value of this given subset and is also assigned, in a statistical way, to those attributes appearing in the subset. Statistically good attributes have a better chance of being selected into an attribute subset of the next generation. This process ends when a satisfactory subset is found or the maximal number of generations is reached. Preliminary experiments done in attribute selection for interpretation of natural scene images (Michalski et al., 1996) show a pronounced speeding of GA-based search for a good attribute subset.

2 Background

Attributes are the basis for man or machine to interpret, classify or recognize scenes or objects from images. However, it is known that due to possible nonlinear interaction among attributes, not any set of attributes can produce the same good results. Thus, attribute selection is needed whose goal is selecting the best subset or a satisfactory one according to some criterion. Image interpretation is a good application domain for attribute selection as there usually exist many numerical attributes and the amount of data is huge, and it is hard to find out, at a glance or by a simple computation, which attribute subset could lead to better results.

A question arises: what is the meaning of a "good" attribute? We observe that there are two kinds of "good" attributes: individually good and collectively good. An attribute is individually good if it itself satisfies some requirements based on analysis of its properties and given data. For example, orthogonality is such a requirement which indicates an attribute's property of measuring different aspects of data. Another example of such requirements is separability which indicates an attribute's ability to separate different object classes. However, there is one problem with selecting attributes directly according to these requirements: though such a set of attributes could lead to satisfactory results, it is not guaranteed because of the nonlinear interaction among attributes. The other problem is that requirements such as orthogonality are rarely operational even by a human being, and requirements such as separability seem to be operational but cannot be determined in reality because of the large number of object classes, and the huge amount of data and noise within the data. An attribute is collectively good if the set of attributes in which this attribute appears brings good results to a system. This concept of "good" captures the cooperation and nonlinear interaction among attributes and is exactly the goal of socalled attribute selection.

The relationship between this two kinds of good attributes is: a satisfactory attribute subset usually contains some individually good attributes; in other words, individually good attributes can be collectively good; a collectively good attribute is not necessarily individually good. The reason for us to introduce the concepts of individually good and collectively good is that their relationship has not drawn enough attention and been utilized for attribute selection.

A naive method for attribute selection is to generate each possible subset of attributes and then test the system performance. However, this method is almost never used in reality because its time cost is exponential with regard to the number of attributes. Another method is ranking candidate attributes based on some criterion followed by deleting some attributes with lower ranks (e.g., Baim, 1982). This method can quickly determine an attribute subset, but it ignores the possible nonlinear interactions among attributes. Some researchers (e.g., Imam and Vafaie, 1994) use heuristic search in attribute selection. This method usually runs fast; however, it could end up with a locally optimal attribute subset. Like the ranking method, this approach is unable to capture nonlinear interaction among attributes and moreover when the number of attributes is large, it is hard or impossible to find effective heuristics that can be used to guide a search process. Forsburg (1976) used an adaptive random search method which increased the probabilities of being selected of those attributes which appeared in generated knowledge descriptions (i.e., they are *relevant* attributes) to make them more likely be selected in the next subset.

Researchers also use genetic algorithms to address the issue of attribute selection (e.g., Vafaie and DeJong, 1993; Bala et al., 1995). This approach utilizes the explorative power of genetic algorithms and searches for a satisfactory attribute subset which captures nonlinear attribute interaction in some degree without exhaustive search. This method can produce good results, especially in the case of a large number of attributes. The work by Bala et al. (1995) is a good example. However, previous work adopting this approach has mainly focused on how to represent a problem so that a genetic algorithm could run on it. It should be pointed out that such a GA-based approach normally consumes much time in genetic evolution, and it becomes even worse when the number of attributes is large, the amount of data is huge and determining a fitness value needs some time. This is often the case in image interpretation. This paper addresses this problem by introducing attribute quality measure into genetic evolution.

3. Methodology

3.1 Attribute Quality Measure

The above observation of two kinds of attribute goodness and their relationship is the basis of this paper. The best or a satisfactory attribute subset usually cannot exist without containing individually good attributes. So during genetic evolution, it may be better to let individually good attributes be more likely to appear in generations. Based on this idea, we try to determine individually good attributes of appearing in an individual of a generation. In contrast, previous work treated each attribute in an equally fair way. Though in this way the explorative power could be strong, many individuals (i.e., attribute subsets) in a generation contain few or no individually good attributes,

and so the whole evolution is likely to consume much time before generating a subset meeting some acceptability criterion.

There are two situations in which attribute quality measure can be introduced into GA evolution: forming the first generation and mutating within one individual of a generation during evolution. We refer to these two sorts of quality measure as *static* and *dynamic* respectively in this paper.

Static quality measure is acquired from analysis of attributes based on given data before evolution. Separability is one property which can be evaluated to some degree from given data and we therefore use it for static quality measure. Other properties such as orthogonality cannot be computed directly from data and therefore must be determined by the designer; thus, they are not considered here. For each attribute, we try to evaluate its separability and assign a value to it. Attributes with high values are considered to be individually good. We increase their probabilities of appearing in the individuals (attribute subset) of the first generation. Notice that these attributes are not guaranteed to appear in the individuals of the first generation.

Static quality measure: C is a set of numbers representing classes. For an attribute Ak ($1 \le k \le N$, where N is the number of candidate attributes), calculate the mean and standard deviation of each class i ($i \in C$) in this attribute from given data (or sampled data), say $\overline{x_{ik}}$ and σ_{ik} . Then Ak's static quality measure is defined as

$$\sum_{\{i,j\}\subseteq C;i< j} \frac{2}{\pi} \tan^{-1} \left[\frac{(\overline{x_{ik}} - \overline{x_{jk}})^2}{1000\sigma_{ik} * \sigma_{jk} + \varepsilon} \right].$$

In the above, ε is a very small number preventing the denominator from being zero. Note that this formula is designed for numerical attributes (in our experiments, all attributes are numerical). For nominal attributes, the methods such as PROMISE (Baim, 1982) are suitable.

Dynamic quality measure is calculated during the evolution process. If an attribute subset, i.e., an individual in a generation, results in good results, then every attribute in this subset will get some credit. If on the average one attribute has a high credit value, then its probability of surviving the mutation so as to appear in a new individual in the next generation is increased. This credit information is used as dynamic quality measure and tries to capture largely the concept of *individually good* and to some degree the concept of *collectively good*.

Dynamic quality measure: For an attribute Ak, add the credit values of all the previous individuals (i.e., attribute subsets) since the first generation in which Ak appeared and divide this sum by the number of such individuals. The result is defined as the dynamic quality measure of attribute Ak.

In the above, the credit value is problem-dependent. In this paper, we use classification accuracy on testing data as an individual's fitness value and also as its attributes' credit value.

In fact, there are many possible ways of defining static or dynamic quality measure for attributes, if reasonable. For example, PROMISE [Baim, 1982] can be used for evaluating static quality measure. The key is to let promising attributes have higher probability of appearing in individuals in upcoming generations.

3.2 GA-Based Attribute Selection

An individual in a generation could be considered as a string of 1s and 0s, in which 1 indicates the attribute is used in this individual and 0 not. The genetic evolution proceeds according to the following steps:

- Step 1: Select the top t_1 attributes according to static quality measure.
- Step 2: Increase their probabilities of appearing in individuals of the first generation by δ_1 .
- Step 3: Randomly select attributes to generate individuals of the first generation.
- Step 4: For each individual in a generation, determine its fitness value and the credit value of each attribute used in this individual.
- Step 5: If some termination criterion is satisfied, then output the satisfactory attribute subset and stop.
- Step 6: Determine each attribute's dynamic quality measure.
- Step 7: Do fitness proportional selection and uniform crossover to generate new individuals for the next generation.
- Step 8: Mutate within each new individual. Two ways: (1) standard mutation, i.e., every attribute has 1/N probability of being mutated (N is the number of candidate attributes); (2) dynamic quality measure enhanced mutation: select the top t_2 attributes according to dynamic quality measure and increase their survival probabilities by δ_2 .
- Step 9: Go to Step 4 to continue the evolution.

In step 3, each attribute can be assigned a value between 0.0 and 1.0 according to a uniform distribution over [0.0, 1.0]. The threshold of selecting an attribute for an individual can be 0.5 and so each attribute is selected with probability 0.5. However, the probabilities of those selected top t_1 attributes should be increased (our experiments used a probability of 0.75) in order that they can more likely appear in the first generation. In step 8, an attribute usually has a probability of 0.5 being mutated. In this work, for each of the selected top t_2 attributes, its probability of being mutated is decreased (0.3 in our experiments) if it already appears in the individual and is increased (0.7 in our experiments) if it does not appear in this individual. The evolution stops when a predefined number of generations are produced, or a termination criterion is satisfied.

4 Example

4.1 Natural Scene Interpretation

We applied a combination of a genetic algorithm and the inductive learning program AQ15c to attribute selection in natural scene interpretation (Michalski et al., 1996),

in which the system is asked to label the class of each area in a natural scene image (see Fig. 1).

4.2 Attribute Definitions

Each pixel in a image is taken as the pixel of interest and a set of attributes are extracted for it. A total of 17 attributes were used in experiments. The first nine are computed according to properties of the pixel itself: (1) red value; (2) green value; (3) blue value; (4) intensity; (5) saturation; (6) hue; (7) relative red value = red - min(red, green, blue); (8) relative green value = green - min(red, green, blue); (9) relative blue value = blue - min(red, green, blue).



Fig. 1. A natural scene image.

-1 0 2 0 -1	-4 0 8 0 -4	-6 0 12 0 -6	-4 0 8 0 -4	-1 0 2 0 -1	-1 -4 -6 -4 -1	0 0 0 0	2 8 12 8 2	0 0 0 0	-1 -4 -6 -4 -1			
(10)) horize	ontal lin	e opera	tor	(11) vertical line operator							
-1 -2 0 2 1	-4 -8 0 8 4	-6 -12 0 12 6	-4 -8 0 8 4	-1 -2 0 2 1	-1 -4 -6 -4 -1	-2 -8 -12 -8 -2	0 0 0 0	2 8 12 8 2	1 4 6 4 1			
(12) horizo	ontal edg	ge opera	ator	(13) vertical edge operator							
-1 -2 0 2 1	0 0 0 0 0	2 4 0 -4 -2	0 0 0 0	-1 -2 0 2 1	-1 0 2 0 -1	-2 0 4 0 -2	0 0 0 0	2 0 -4 0 2	1 0 2 0 1			
(14)	horizon	tal V-sh	ape ope	erator	(15) vertical V-shape operator							
1 -4 6 -4 1	-4 -24 16 -4	6 -24 36 -24 6	-4 -24 16 -4	1 -4 6 -4 1		1 -2 1	-2 4 -2	1 -2 1				
(1	6) frequ	ency sp	ot oper:	ator	(17) Laplacian operator							

Fig. 2, Laws' energy filters for generating attributes.

The other eight attributes are computed according to Laws' energy filters in Fig. 2, which detect information such as directionality and roughness around the pixel of interest (Laws, 1980). The usage of each matrix is such: let the center of a matrix positioned at the pixel of interest, multiply each value in the matrix by the gray value of the pixel in the corresponding position, sum all the products to get the attribute value.

4.3 Data and Discretization

A 20 x 20 area of each class within the natural scene is selected from Fig. 1 (boxes) and 17 attributes are computed for each pixel in the area. Fig. 3 gives some examples. 60% of all the selected data are randomly taken for learning and the other 40% for testing (Weiss and Kulikowski, 1992). Note that before learning rules to describe pixels by using AQ15c, the value of each attribute is uniformly discretized to one of fifteen levels for the experiments in this paper. Actually any discretization scheme is applicable here. A pixel description (i.e. rule) learned by AQ15c is exemplified in Fig. 4.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17
6	6	5	6	2	6	1	0	0	5	5	5	3	2	9	1	5
8	7	6	7	2	6	2	1	0	6	6	5	5	7	9	2	6

Fig. 3. 17 attribute values of 2 selected rock pixels.

Rock <:: [A1=5..14] [A5=0..4] [A13=3..10] [A14=1..7] [A15=9..12]

Fig. 4. One of the learned rules of rock pixels.

4.4 Static Attribute Quality Measure

Static quality measure is calculated for every attribute based on randomly selected training data. See Fig. 5.

A1 A2 A3 A4 A5 A6 A7 A8 A9 8.390 5.324 8.328 4.769 5.116 11.122 9.591 12.456 5.000 A10 A11 A12 A13 A14 A15 A16 A17 6.661 8.912 8.199 7.536 8.075 6.582 8.967 3.957

Fig. 5 Static attribute information

From our experience, static quality measure for attributes like hue, and relative red and green values (A6, A7, A8 respectively) should be high, and the above values reflect this.

5 Experiments and Discussion

5.1 Experimental Results

The genetic algorithm in De Jong (1996) was adopted. The population size was 20 and the experiments were done on a Sparc 2 workstation.

Three kinds of experiments were done: "traditional" refers to the way described in (Bala et. al., 1995); "static" means only static quality measure is introduced into genetic evolution; "static+dynamic" means both static and dynamic quality measure are used. 10 runs were performed for each kind of experiment. The average testing

accuracy is plotted as best-so-far in the Y axis against the number of births in the X axis (Fig. 6).

5.2 Discussion

Fig. 6 shows a significant speeding effect due to introduced attribute quality measure. Dynamic quality measure did not result in much improvement at the early phase of evolution, because a few generations are not able to capture the statistical goodness of each attribute. When there were enough generations created, dynamic quality measure worked to some degree. Note that both static and dynamic quality measures had a strong positive effect on evolution speed. It is possible to gain better performance if we do not give credit to each attribute in an individual during evolution but rather only these attributes which were actually used in obtained knowledge descriptions because only they contributed to acquired testing accuracy (Forsburg, 1976).



Fig. 6. The comparison of evolution speeds. Best-So-Far is testing accuracy.

For the success of attribute quality measure, the design of formulas of quality measure seems crucial. We tried another formula for static quality measure (not shown in this paper) but the speeding effect was not good. We set parameters t_1 and t_2 in the above methodology to 5. We also tried selection of the top 3, 4, 6 attributes for probabilities to be increased. The results showed that the top 5 were the best for this given problem. We consider this issue to be problem-dependent and very important. If a system itself knows too much about the properties of candidate attributes, it could simply select them, and this way would clearly produce very good results quickly. Obviously, this is not always true. It is the case that only some of them may be selected. If too many attributes' probabilities are increased without well-founded understanding of them, then the evolution process is subject to going to and staying at some local optimum or spending more time in finding a satisfactory subset than without attribute quality measure. On the other hand, selecting too few attributes for probability increase is important.

Note that our methodology is similar to the work by Forsburg (1976) but different in many aspects. Attribute quality measure is like *information content value* mentioned there; nonetheless the latter did not touch upon the concept of *static* quality measure. Even though Forsburg adopted a random search, its theoretical properties were unclear. To some degree, the work there can be considered as a special case of our methodology with population size being one (no crossover, no mutation) and thus its search is not so powerful and systematic as genetic algorithms are. Further, we cannot evaluate its performance in terms of accuracy and speeding effect since they were not reported there.

6 Conclusion

This paper describes a promising way of speeding GA-based attribute selection by introducing attribute quality measure. It combines a genetic algorithm and the inductive learning program AQ15c into a multistrategy learning system. Quality measure of each attribute is introduced to determine whether to increase its survival probability. Experimental results are presented to show the feasibility of this methodology. The preliminary results indicate an improvement in time in comparison with previous GA-based work in which the main focus was on representing a problem so that a genetic algorithm could work on it.

There are some aspects which require further work. Among them, selection of an appropriate number of attributes whose probabilities of survival are going to be increased is of special interest. It is desirable to find an adaptive way of determining this number so that the system could run more independently of the system designers and domain experts. Another future experiment is to take as testbed more attributes and more application domains, especially in the case of large number of attributes (only 17 attributes were tested in this paper to show the methodology). Further, it is worthwhile to design other effective formulas to calculate attributes' quality levels, so as to capture more attribute properties.

We believe that in application domains with large quantities of numerical data such as image interpretation, introduction of attribute quality measure is a promising way to speed up GA-based attribute selection.

Acknowledgments

The author would like to thank Dr. Michalski and Dr. Kaufman for their criticism and suggestions to the earlier draft of this paper. This research was conducted in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's activities are supported in part by the Defense Advanced Research Projects Agency under grant F49620-95-1-0462 administered by the Air Force Office of Scientific Research, in part by the National Science Foundation under grants DMI-9496192 and IRI-9020266, and in part by the Office of Naval Research under grant N00014-91-J-1351.

References

- Baim, P.W., 1982. The PROMISE Method for Selecting Most Relevant Attributes for Inductive Learning Systems. *Reports of the Intelligent Systems Group*, Department of Computer Science, University of Illinois at Urbana-Champaign, ISG 82-5.
- Bala, J., Huang, J., Vafaie, H., DeJong, K. and Wechsler, H, 1995. Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, August, Motreal, Quebec, Canada.
- De Jong, K., 1996. Evolutionary Computation: Theory and Practice. MIT Press.
- Forsburg, S., 1976. AQPLUS: An adaptive random search method for selecting a best set of attribute from a large collection of candidates. Internal Report, Department of Computer Science, University of Illinois, Urbana, Illinois.
- Imam, I.F., and Vafaie, H., 1994. An Empirical Comparison Between Global and Greedy-like Search for Attribute selection. *Proceedings of the Seventh Florida Artificial Intelligence Research Symposium*, May, Pensacola Beach, FL., pp. 66-70.
- John, G., Kohavi, R., and Pfleger, K., 1994. Irrelevant Features and the Subset Selection Problem. Proceedings of the Eleventh International Conference on Machine Learning, July, Rutgers University in New Brunswick, New Jersey, pp. 121-129.
- Koller, D., and Sahami, M., 1996. Toward Optimal Feature Selection. *Proceedings* of the Thirteenth International Conference on Machine Learning, July, Bari, Italy, pp. 284-292.
- Kreithen, D.E., Halversen, S.D. and Owirka, F.J., 1993. Discriminating Targets from Clutter. *Journal of MIT Lincoln Lab.*, pp. 25-52.
- Laws, K., 1980. Textured Image Segmentation. Ph.D. Thesis, University of Southern California, Department of Electrical Engineering.
- Michalski, R.S., Mozetic, I., Hong, J., and Lavrac, N., 1986. The Multipurpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. *Proceedings of the Fifth National Conference on Artificial Intelligence*, August, Philadelphia, PA., pp. 1041-1045.
- Michalski, R.S., Zhang, Q., Maloof, M.A., and Bloedorn, E., 1996. The MIST Methodology and Its Application to Natural Scene Interpretation. *Proceedings of Image Understanding Workshop*, Palms Spring, CA., pp. 1473-1479.
- Vafaie, H. and De Jong, K., 1993. Robust Feature Selection Algorithms. *Proceedings of the International Conference on Tools with AI*, Boston, MA., pp. 356-364.
- Weiss, S.M., & Kulikowski, C.A., 1992. Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning and expert systems, Morgan Kaufmann, San Mateo, CA.
- Wnek, J., Kaufman, K., Bloedorn, E., and Michalski, R.S., 1995. Selective Induction and Learning System AQ15c: the Method and User's Guide. *Reports of the Machine Learning and Inference Lab.*, MLI 95-4, Machine Learning and Inference Lab., George Mason University, Fairfax, VA.