Pattern Recognition Via a Genetic Algorithm

Angel Kuri M.
CENTRO DE INVESTIGACION EN COMPUTACION
INSTITUTO POLITECNICO NACIONAL
Blvd. Adolfo López Mateos
Col. Lindavista
México D.F.
729-6000 ext. 54301
akuri@pollux.cenac.ipn.mx

Abstract.

In this paper we describe a heuristic approach to the problem of identifying a pattern embedded within a figure from a predefined set of patterns via the utilization of a genetic algorithm. In part 1 we describe the general process. In part 2 we discuss the type of genetic algorithm utilized. In part 3 we report some results obtained from a test set. In part 4 we discuss the aforementioned results and we comment on these. We also point out some possible extensions and future directions.

General Process.

The problem of pattern recognition has long been considered to be a topic of particular interest in many areas of Computer Science. It has been tackled in many ways throughout a considerably large span of time and, as of today, it remains a subject of continuous study.

In this paper we report yet another method to solve a particular problem of pattern recognition. This problem may be described as follows:

a) We are given a certain graphical figure, possibly composed of an unknown non-linear combination of simpler (component) graphical figures. For our discussion we shall call this figure the master figure or, simply, the master.

b) We are given a set of "candidate" figures which we are interested to discover within the master if any of these does indeed "lie" within it. What we mean when we say that a figure lies within another figure is, in fact, central to our method and we shall dwell upon this matter in what follows. We shall call these possibly embedded figures the patterns.

c) We would like to ascertain, with a given degree of certainty, whether one or several of the patterns lie within the master.

In figure 1, a set of patterns is shown surrounding a master.
The main problem of defining when a pattern \textit{lies} within a master is interesting only when such relation is not direct. That is, it is relatively simple to find, say, a square within a master if such a square is one of the patterns on a 1 to 1 scale, in the same position and with the same inclination. Here, however, we are looking for a much more general case. Namely, we impose upon our method, the task of identifying a pattern in the master even if the pattern is found on a different scale, a different position and a different inclination than the ones in the pattern. We shall define, therefore, three operators on the pattern:

1. A scale operator, which we denote by \( S(s, f) \).

Where \( s \) is the scale factor and \( f \) is the figure being scaled.

2. A rotation operator, which we denote by \( R(r, f) \).

Where \( r \) is the rotation angle, and \( f \) is the figure being rotated.

3. A translation operator which we denote by \( T(t_1, t_2, f) \).

Where \( t_1, t_2 \) are the translations on the x and y axis respectively and \( f \) is the figure being translated.

Henceforth, a pattern is mapped an arbitrarily selected number of times under scaling, rotation and translation into a \textit{derived pattern} or \textit{descendant}. That is, from every pattern we extract a family of descendants which arise from a process of repeated application of \( S, R \) and \( T \).

The rationale behind this strategy is simple: under the assumption that the possible number of configurations is sufficiently large, we settle with a sample of the configuration space in the hope of capturing the essence of the pattern in all its possible states where by "state" we mean any one of the possible combinations of \( s, r, t_1, t_2 \). Clearly the number of possible states is sufficiently large so that an exhaustive enumeration of these is impractical. The size of the sample is largely determined, therefore, by practical considerations such as the amount of information contained in the figure, the speed of the computer where the process of identification is performed and the amount of memory at our disposal. We denote the family of descendants for a given figure \( f \) by \( \mathcal{F}(f) \).

Once the samples (one per pattern) are obtained, we attempt to characterize the relationship between the \( i \)-th pattern and the master by minimizing a norm which should reflect the distance between the master and the pattern. If the said distance is minimum we shall accept the fact that the pattern is embedded within the master. This is what we mean when we say that a pattern \textit{lies} within a figure.

That is, we shall accept that pattern \( f \) is found in the master \( m \) if the distance between \( f \) and \( m \) is relatively small.

We could have, of course, attempted to minimize such distance from a "traditional" norm such as \( L_1 \), \( L_2 \) of \( L_\infty \). Indeed, such a scheme was applied in (1) where we attempted to minimize the said distance in \( L_\infty \). There we recognized a set of fuzzy alphabetic (i.e. A, B, ..., Z, ...) characters with a neural network and with a scheme similar to the one reported here. There, however, the patterns were unique.
That is, the master consisted of only one of the fuzzified characters and the method performed unsatisfactorily when several patterns were put together. Our conclusion was that the problem lied in the fact that \( \Phi(f) 2 \) was not sufficiently rich and/or the master was too complex.

In order to overcome the limitations outlined above we felt it necessary to enrich \( \Phi 3 \) and to adopt an ad hoc distance norm. To achieve these two goals while retaining the essence of the method we appealed to a genetic algorithm.

In the algorithm a set of random patterns (each of which will act as a probe \( \pi 4 \)) is generated. Thus, information is maximized to begin with. That is, in terms of a Fourier decomposition no harmonics of the pattern are left out. Then the distance between the test pattern and both \( \Phi 5 \) and the master is minimized simultaneously. To do this:

\[ d_{\Delta} = \frac{I}{N} \sum (\pi - \delta_i) 8. \]

\[ m_{\Delta} = -m_{\Delta_{\pi}}. \]

\[ d_{\Delta_{\pi}} = \frac{1}{2}(\Delta_{\delta} + \Delta_{m}). \]

The genetic algorithm receives as its initial population, therefore, a set of random patterns. Its fitness function is then the mutual distance \( d_{\Delta_{\pi}} 10 \) which it tries to minimize. The population, thereafter, evolves to a fittest individual which is constantly closer to both the set of descendants and the master, thereby establishing an informationally meaningful link between the particular sample from which \( \Phi(f) 11 \) originally arose and the master.

In order to calculate the above mentioned distances the system is fed a set of figures in compressed (PCX) format. This set comprises both the master and the patterns. As a second step, the \( \delta_{12} \) are generated. Once having these samples (the arguments of the operators are generated randomly) the genetic string of the master and the descendants is composed of the a concatenation of the rows of the pattern (1 bit per pixel). The distance between probe and descendants, on the one hand, and probe and master, on the other, is trivially calculated by counting those positions where the values agree. In our test we only considered black and white images. Therefore, an empty space is a "1", whereas a filled space is a "0". It should be clear that the genetic algorithm is basically measuring the information content coincidence in the probe vs. the descendants and vs. the master. It should also be clear that the fact that pattern information is state independent given a properly selected size of the sample. That is, regardless of where the pattern may be within the space of the master and regardless of its position and/or angle of inclination relative to the master, as long as the information of the pattern remains a match will be found, that is, as long as the pattern is not deformed.

**Genetic Algorithm.**

For those familiar with the methodology of genetic algorithms it should come as no surprise that a number of questions relative to the best operation of the algorithm immediately arose. The Simple Genetic Algorithm frequently mentioned in the literature leaves open the optimal values of, at least, the following parameters (2,3,4):

a) Probability of crossover (\( P_c 13 \)).
b) Probability of mutation (\( P_m 14 \)).
c) Population size.

Additionally, premature and/or slow convergence are also of prime importance (5,6). In the past we have conducted experiments (7,8,9) which led us to take the following decisions:

a) We utilize the Vasconcelos scheme, i.e. selection is deterministic on an extremum crossover schedule with N-elitism.
b) Crossover is annular.
c) \( P_c, P_m \) and population size are self-adaptive.
d) A Random Mutation Hill Climbing algorithm was included to accelerate the expected rate of convergence.

In what follows we describe in more detail the points just outlined.

**An "Optimal" Genetic Algorithm.**

Optimality in GA's depends on the model chosen. For reasons beyond the scope of this paper (but see references 10, 11, 12) we have chosen to incorporate in our model the following features:

1. Initial population.
   It was randomly generated. We decided not to bias the initial population in any sense for the reasons outlined above. For these work we selected initial populations of size 50.
2. Elitism.
All of the individuals stemming from the genetic process are rated according to their performance. It has been repeatedly shown that elitism leads to faster convergence.

3. Selection.

The individuals are selected deterministically. The best (overall) N individuals are considered. The best and worst individuals (1-N) are selected; then the second best and next-to-the-worst individuals (2-[N-1]) are selected, etc.

This is known as the Vasconcelos model of GA's. Vasconcelos model has shown to guarantee that there is no premature convergence and that, as a consequence, it reaches generally better results than other models.


It is performed with a probability $P_c$. In our GA $P_c$ is variable and $.9 \leq P_c \leq 1.0$ 14. Further, we adopted annular crossover. Annular crossover makes this operation position independent. In the past several authors (13) have attempted to achieve position independence in the genome. Annular crossover allows for unbiased building block search, a central feature to GA's strength.

5. Mutation.

Mutation is performed with probability $P_m$. It is also variable and $0.005 \leq P_m \leq 0.01$ 15. Mutation is kept at very low levels.


The number of offspring from the process is variable and lies between 2 and three per cross. Once an individual is selected for crossover its two descendants are chosen at least once and at most thrice.

7. Hill Climbing.

Our GA is enriched by incorporating a Random Mutation Hill Climber (RMHC). It is generally recognized (14) that hill climbing algorithms tend to somewhat enhance the search capacities and are, therefore, sometimes applied after the GA has performed. Here the a RMHC is incorporated as a part of the GA itself. Every time the RMHC is invoked, it is allowed to generate a set of individuals whose evaluation cost ($\Xi$ 16) should not exceed a maximum relative to the overall process. Here $1\% \leq \Xi \leq 5\%$ 17. The RMHC is invoked with probability $P_H$. $P_H$ is variable and $1 \leq P_H \leq 18$.

Self Adaptation.

In past works we have reported on the possibility and convenience of modifying the behavior of the GA since the operating needs vary over the solution of the problem. The solution we adopted in the past and which we use here is to incorporate the parameters of the GA as part of its own genome. In this case we have included:

a) A 4 bit attribute to represent $P_c$. That is, $P_c$ may take any of 16 possible different values of which we allow only the ones from 1 to 10. We then tacitly divide by 10.

b) A 4 bit attribute to represent $P_m$. Here we restrict the set of values to the ones from 1 to 5. We tacitly divide by 100.

c) A 4 bit attribute to set the number of offspring $\eta$. This may range from 1 to 5. The set of values is adjusted accordingly.

d) A 4 bit attribute representing $\Xi$. Its value may range from 0 to 16 and it is tacitly divided by 100. Its average value is an input parameter for the system. As mentioned before, we set its value between 1 and 5%.

e) A 1 bit attribute to mark an individual so that it is possible to determine whether such individual arose from the genetic (selection-reproduction-crossover-mutation [SRCM]) scheme or from RMHC.

In every step of the process a quartet of genetic parameters ($P_c$, $P_m$, $\eta_G$, and $\Xi_G$ ) is evaluated.

$$P_c = \frac{1}{N} \sum_{i=1}^{N} (P_c)_i$$

$$P_m = \frac{1}{N} \sum_{i=1}^{N} (P_m)_i$$

$$\eta_G = \frac{1}{N} \sum_{i=1}^{N} \eta_i$$

$$\Xi_G = \frac{1}{N} \sum_{i=1}^{N} \Xi_i$$
These parameters act upon each of the individuals of the population for the duration of the present generation only. That is, the more successful individuals determine how the genetic process is to proceed from the behavior of the historically best fit segment of the population taken as a whole. Since these parameters are calculated for each generation, they evolve along with the best individuals.

Also, the decision to invoke either the SRCM or the RMHC is taken. \( P_H \), the probability of invoking the RMHC is calculated:

\[
P_H = \frac{1}{N} \sum_{i=1}^{N} (P_H)_i \quad \text{(6.1)}
\]

If \( P_H > \frac{1}{2} \) then \( P_H = \frac{1}{2} \) \quad \text{(6.2)}

If \( P_H < \frac{1}{8} \) then \( P_H = \frac{1}{8} \) \quad \text{(6.3)}

A random number \( \Psi \) is generated. If \( \Psi \geq P_H \) then RMHC is invoked. Otherwise, SRCM is invoked.

The RMHC is, therefore, an occasional actor which performs frequently if its offspring are majority (but never more frequently than the genetic process itself) and infrequently when they are minority (but never less than \( \frac{1}{8} \) of the time).

All of the attributes mentioned form part of the genome and are subject to the genetic operators of crossover and mutation. Of course, being part of the genome, they undergo all of the transformations from the overall genetic conduct described above.

The genome is, thus, formed of two operationally distinct parts:

a) A pattern representation.
b) A genetic representation.

The pattern representation consisted, in our case, of a 5625 bit long binary string.

The genetic representation consisted of 17 bits as described above.

Additionally, each individual's genome held the fitness value for such individual.

The scheme of the full genome is shown in Figure 2.

<table>
<thead>
<tr>
<th>( P )</th>
<th>( P_m )</th>
<th>( \eta )</th>
<th>( \Xi )</th>
<th>Genetic/Hillclimber</th>
<th>Fitness</th>
<th>Genome</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 bits</td>
<td>4 bits</td>
<td>4 bits</td>
<td>4 bits</td>
<td>1 bit</td>
<td>(5625 bits)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.** The Genome for the Genetic Algorithm.

**Results.**

We performed an initial set of experiments to test the scheme we have described. We designed 5 sets of 10 descendants each (i.e. \( \Phi_1(f_1), \Phi_1(f_2), \Phi_1(f_3), \Phi_1(f_4), \Phi_1(f_5) \)) and 5 sets of 25 descendants each (i.e. \( \Phi_2(f_1), \Phi_2(f_2), \Phi_2(f_3), \Phi_2(f_4), \Phi_2(f_5) \)). The descendants were obtained by applying operators \( \sigma \) and \( \tau \) as described above. These we matched vs. a master figure (the patterns and the master figure correspond to the ones shown in Figure 1.).

The two following tables show the results for these experiments.

The selection criterium is actually very simple: accept as an embedded pattern the one which shows the smallest \( \Delta_{m} \).
<table>
<thead>
<tr>
<th>Pattern</th>
<th>Maximum</th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Selected Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>170.70</td>
<td>169.60</td>
<td>.37</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>140.90</td>
<td>139.39</td>
<td>.39</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>173.8</td>
<td>172.89</td>
<td>.35</td>
<td>YES</td>
</tr>
<tr>
<td>4</td>
<td>150.70</td>
<td>150.24</td>
<td>.09</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>136.40</td>
<td>135.29</td>
<td>.36</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 1. Results for $\Phi_{23}$

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Maximum</th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Selected Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>180.17</td>
<td>179.50</td>
<td>.35</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>150.90</td>
<td>148.39</td>
<td>.42</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>185.7</td>
<td>187.32</td>
<td>.35</td>
<td>YES</td>
</tr>
<tr>
<td>4</td>
<td>156.70</td>
<td>155.00</td>
<td>.12</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>137.44</td>
<td>136.79</td>
<td>.38</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 2. Results for $\Phi_{24}$

In the tables above, the smallest $\Delta_{\text{min}}$ correspond to patterns 1 and 3, which are shown in Figure 3. The patterns not recognized are shown in figure 4. The master figure is shown in Figure 5.

**Figure 3.** Patterns 1 and 3.
As seen, in this simple trial the correlation between the matches and what our intuition would dictate is accurate. This results are encouraging on two accounts:

First, it would seem that the method, in general, should be expected to yield reasonably good results.
Second, from the tables above, it seems that whenever $\Phi(f)$ 26 is enriched, the precision is enhanced.

Conclusions.

The method described above is a hybrid of what has been, in the past, called supervised and unsupervised learning. We quote: "The need for learning arises whenever available a priori information is incomplete. The type of learning depends on the degree of completeness of this a priori information. In learning with supervision, it is assumed that at each instant of time we know in advance the desired response of the learning system, and we use the difference between the desired and actual response, that is, the error of the learning system, to correct its behavior. In learning without supervision, we do not know the desired response of the learning system". (15).

Here, the genetic algorithm, by its own definition evolves under supervision of the environment, via the fitness function. However, the fitness function itself has been determined by a random sampling of the possible space of solutions. In that sense the search is unsupervised, in as much as the way the spectrum of the restricted sample space is "blind".
We believe this blend of supervised-unsupervised learning is what gives the proposed method interest. Of course, much work remains to be done.

In the first place, the patterns and the master were specifically selected to validate the model in first instance. The figures that we attempted to recognize were directly embedded in the master. What would happen if the sought for relationship were not so straightforward? We intend to pursue further investigation along these lines.

Secondly, although the results of the model, as reported, correspond to intuition, it is not clear that this intuition could be substantiated in general. For example, are we sure that the figure in pattern 4, for instance, is not informationally contained in the master?

Thirdly, the measure that we used is amenable to reconsideration and enrichment. Do we have certainty that this direct difference is the best norm? We remarked earlier that a way to ensure that this norm is adequate is that the algorithm itself selects it. This is true in the sense that the genetic methodology assigns a certain degree of fuzziness to the measure. For example, the random walk implicit (14) in the GA translates into the fact that the distance being minimized is adaptive. Similar distances (but not identical) would have been obtained by slight changes in the initial conditions set in the GA.

Finally, the technical details (not glossed on in the paper) were quite interesting in themselves. The mere process of applying the defined operators $\sigma, \rho$ and $\tau$ became an interesting matter. The figures were to remain within the boundaries of the master's environment; the discreetness of the binary mapping disallowed certain arguments $s, r$ and $t_1, t_2$; the repeated comparison of the probe vs. the patterns in $\Phi(f)$ implied vast amounts of computation.

In view of the technical shortcomings of the method, it remains to generalize it and prove its universality before attempting to solve large practical problems with it. We hope to report on the (successful) completion of the investigation shortly.
References.