We implement genetic algorithm based predictive model building as an alternative to the traditional stepwise regression. We then employ the Information Complexity Measure (ICOMP) as a measure of model fitness instead of the commonly used measure of $R^2$-square. Furthermore, we propose some modifications to the genetic algorithm to increase the overall efficiency.

Keywords: genetic algorithms, information complexity measure, stepwise regression, diversification, population reduction
where $L(\hat{\theta}_k)$ denotes the maximum likelihood function, $\hat{\theta}_k$ is the maximum likelihood estimate of parameter vector $\theta$, and $m(k)$ is the number of parameters in the model. The first term of AIC gives the lack of fit of the model, and the second term is a penalty for the number of parameters in the model. The model with the lowest AIC value is considered the best, because the model successfully determines the underlying stochastic process with the least number of parameters. Although AIC does take into account the problem of over-fitting, where other measures such as R-square do not, AIC is not sensitive to parameter dependency, which is an important component for model selection. If a model with both low variance and low covariance can be produced, then the parameters can be better estimated, as they will not be correlated. As an alternative to AIC, we consider ICOMP as a complexity measure which considers variance and covariance, and accounts for the problem of over-fitting the model. It is calculated by

$$ICOMP = - \log L(\hat{\theta}_k) + 2C(\hat{\Sigma}_{model}),$$

where $L(\hat{\theta}_k)$ again denotes the maximum likelihood function, $\hat{\theta}_k$ is the maximum likelihood estimate of parameter vector $\theta$, under the model $m_\gamma$, $C$ is a real-valued complexity measure, and $\hat{\Sigma}_{model}$ is the estimated covariance matrix of the parameters of the model. Note that the first term in AIC is double the first term in ICOMP. The main difference between the two measures of complexity is that AIC only considers the number of parameters as a penalty, whereas ICOMP considers the covariance between parameters. In predictive model building, we use ICOMP (IFIM) as our multiple regression model selection criterion. This value for ICOMP is based on the inverse-Fisher information matrix (IFIM). For multiple regression, the value of ICOMP can be directly calculated after regression is implemented, and is given by

$$ICOMP(IFIM)_{Regression} = n \log(2\pi) + n \log(\hat{\sigma}^2) + n + C_1 \left( \hat{F}^{-1}(\hat{\theta}_m) \right)$$

where

$$C_1 \left( \hat{F}^{-1}(\hat{\theta}_m) \right) = (q + 1) \log \left[ \frac{\text{tr} \left( \hat{\sigma}^2(XX')^{-1} \right)}{q + 1} \right] - \log \left| \hat{\sigma}^2(XX')^{-1} \right| - \log \left| \hat{\sigma}^2 \right|$$

and $n$ is the number of parameters in the model, $q$ is the number of observations, $\hat{\sigma}^2 = SSE/n$, $\text{tr} \left( \hat{\sigma}^2(XX')^{-1} \right)$ is the trace of the observation matrix multiplied by its inverse and then scaled by $\hat{\sigma}^2$, and $\left| \hat{\sigma}^2(XX')^{-1} \right|$ is the determinant of the previous matrix.

Since the model with the lowest ICOMP value is considered the best, the GA chooses strings biased toward those with the lowest value. A commonly used method to form the mating pool is “proportional selection,” which depends on selecting strings for the mating pool with a probability proportional to their fitnesses. In proportional selection, the first step of the calculation of the fitness values is subtracting the ICOMP value of each string in that generation from the maximum value of ICOMP in the population. That is

$$\Delta ICOPM(IFIM)_{(i)} = ICOPM(IFIM)_{(\text{Max})} - ICOPM(IFIM)_{(i)}$$

for each $i = 1,2,\ldots,N$, where $N$ is the size of the population. Then the average ICOMP difference (the “average fitness”) for the total population is calculated as

$$\Delta ICOPM(IFIM) = \frac{1}{N} \sum_{i=1}^{N} \Delta ICOPM(IFIM)_{(i)}$$

Finally, each string is given a fitness value that is the ratio of its ICOMP difference and the average fitness of the population:

$$\text{Fitness}_{(i)} = \frac{\Delta ICOPM(IFIM)_{(i)}}{\Delta ICOPM(IFIM)}$$

A GENETIC ALGORITHM FOR MULTIPLE LINEAR REGRESSION MODEL SELECTION

Here we consider the implementation of GA’s for predictive model selection and discuss possible improvements.

BACKGROUND

The first step to implementing a GA for any optimization problem is to encode the input variable into binary strings. In the case of multiple linear regression, we have $q$ data points with $n$ explanatory variables and one response variable. We wish to fit the data to

$$y = X\beta + \varepsilon,$$

where $y$ is an $n \times 1$ response vector, $X$ is an $n \times q$ matrix of the data points, $\beta$ is a $q \times 1$ coefficient matrix, and $\varepsilon$ is an $n \times 1$ error vector with entries from independent normal distributions $[N(0, \sigma^2)$ for all components]. The encoding is done by creating a binary string which has $n + 1$ bits, where each bit represents a different parameter of the model and an intercept. The last $n$ bits correspond to the $n$ explanatory variables contained in the dataset, whereas the first bit is the intercept for the linear model. A parameter is included in the model if the value of the bit for that parameter is a 1 and is excluded if it is a 0. For example, suppose we have a dataset where we are interested in predicting the reproductive fitness of a species of trees. The possible explanatory variables may include:

1. Age of tree,
2. Height of tree,
3. Soil pH,
4. Density of trees in the surrounding area,
5. Average temperature of environment,
6. Average rainfall of environment,
7. Circumference of trunk,
8. Longitude of environment,
9. Latitude of environment,
10. Prevalence of disease in environment.
In this case, each binary string will have 11 bits. For example, the string 10010111101 would represent a model which includes the intercept, soil pH, average temperature of environment, average rainfall of environment, circumference of trunk, longitude of environment, and prevalence of disease in environment. Similarly, the string 00001000010 is a model that has no intercept, and includes density of trees in the surrounding area, longitude of environment, and latitude of environment (see Table 1).

The probability that a string will be chosen for the mating pool is proportional to its fitness value. Note that the string with the worst ICOMP value will never be picked for the mating pool, as its fitness will be 0.

Now that we have a method of encoding information and a method to evaluate the fitness values, we have to determine the remaining parameters of the GA. The first one we consider is the method of creating the initial population and determining its size. Unless previous knowledge about the problem is given, it is commonplace in GAs to randomly generate binary strings (Goldberg, 1989). However, in the case of model selection, a user may want to force a parameter(s) to be included, even if it is not part of the model with the lowest complexity. In this case, the initial population can be generated in such a way that certain parameters are always in the model. In addition to determining the method to generate the population, the user must choose the size of the initial population. This decision can be difficult. Generally, the size should not be too large, as this will slow the algorithm, and should not be so small that genetic drift takes over the course of evolution of the population. In typical GAs, the size of the population stays the same; however, this may not be an effective use of computation. We will see in the next section that starting with a larger size may reduce the use of computation may be more effective.

Finally, we discuss the genetic operators which allow the algorithm to find the optimal model. There are two operators that are generally implemented in GAs: crossover and mutation. Crossover mimics biological crossover in a simplified manner. First, the probability of crossover, \( p_c \), is chosen. In the mating pool, a pair of strings are chosen along with a random number from \([0, 1]\). If that number is less than the probability of crossover, then that pair of strings will be crossed, and if \( p_c = 0 \), then no strings will be altered by crossover. After the choice of \( p_c \), the number of crossover points must be chosen. The location of the crossover points is chosen at random. Then the bits from the parent strings are swapped to create two new offspring strings (see Figure 1). The purpose of crossover is to bring together models which have components that reduce complexity. Recall the previous example about trees, where we specified two strings, which we will call Parent 1 and Parent 2. Applying crossover to the two parents creates two offspring (see Figure 1), where Offspring 1 represents a model with an intercept, soil pH, average temperature of environment, longitude of environment, latitude of environment, and prevalence of disease in environment, and Offspring 2 represents a model that includes density of trees in the surrounding area, average rainfall of environment, circumference of trunk, and longitude of environment. Through successive generations and application of crossover of low complexity models, the algorithm is able to find the least complex model (or something close to it) to explain the data.

Crossover can only generate models that include parameters which already exist in the population. But, what happens if the actual least complex model includes a parameter that is not present in the population, that is, the position in the string that represents the parameter is fixed at 0? Mutation alleviates this problem. Mutation in GAs is similar to the mutation that occurs naturally in DNA. First, the probability of mutation, \( p_m \), has to be determined. This value gives the probability that at each location in the string the bit will be flipped. Flipping is defined as a change of a 0 to 1 or a 1 to a 0. Typically, mutation rates are low, on the order of \(10^{-3} \) to \(10^{-5}\). However, strings used for other applications of GAs are usually longer than the ones used for determining least complex models. Although there are ongoing studies on determining optimal crossover and mutation rates (such as Nested GAs, Self-adjusting parameterless GAs), these rates can be determined by trial and error or by pilot runs before the actual data set has been used to build a model.

We conclude this section with a pseudo code for a GA used to find the least complex model that sufficiently describes the data.

1. Generate Initial Population
2. While (\( t < \) Max Generations OR the maximum number of computations have not been executed)
   (a) Calculate ICOMP for the model each string encodes
   (b) Select strings for the mating pool
   (c) Create a new population using crossover
   (d) Mutate new population
   (e) \( t = t + 1 \)
3. End

**MODEL BUILDING VIA ACCELERATED GENETIC ALGORITHMS**

While the use of a typical GA for model selection already proves to be more efficient than stepwise regression, with a few modifications, the process can show a 10-fold increase in accuracy given the same

---

**Table 1 | Chromosomes and variables included by the model it represents.**

| Chromosome    | Variables included          |
|---------------|----------------------------|
| 10010111101   | Intercept, 3, 6, 7, 8, 10  |
| 00001000110   | 4, 8, 9                    |

**FIGURE 1 | Diagram of crossover with two points.**
The change is calculated by the formula

\[
\Delta f_t = f_t^{\text{best}} - f_{t+1}^{\text{best}}
\]

where \( f_t^{\text{best}} \) and \( f_{t+1}^{\text{best}} \) represent the fitness values of the best chromosomes at generation \( t \) and \( t+1 \) respectively. The limit is denoted by \( \Delta f_{\text{max}} \). The percent change in fitness at generation \( t \) can be calculated as

\[
\frac{\Delta f_t}{f_t^{\text{best}}} \times 100
\]

This limit on the reduction may be determined by pilot studies. While model selection remains to be a difficult procedure in case of multiple regressions, using a GA to find the least complex model was found. Testing was done to ensure the same ICOMP values were being generated for the MATLAB and Java code. We performed comparisons between Bozdogan’s original setup and four cases with our modifications. These cases differed in the value of \( \Delta f_{\text{max}} \), and as a result in the initial population size.

Table 2 gives the parameters that were the same for all different setups. Each different GA scheme ran through 200 trials and the number of times the correct model was selected was recorded. Table 3 gives the results.

**CONCLUSION**

While model selection remains to be a difficult procedure in case of a large number of parameters, using a GA to find the least complex model can be quite helpful. We have shown that our modifications to the original GA for model selection can yield strong results.
Additionally, the GA approach (because of the use of ICOMP) is better at handling data in which collinearity exist than the traditional selection methods such as forward, backward, and stepwise selection. In particular it is clear that the modifications had a large effect on the accuracy of the GA. All of the GAs which implemented our modifications significantly outperformed Bozdogan’s GA. This seems to indicate that we may reduce computation and still get statistically the same accuracy if we employ diversification. In all trials, diversification never decreased accuracy. Along with the facts presented above and the fact that diversification is easy (and not costly) to implement, it is our recommendation that it be used for model selection using GAs.

### Table 2 | Parameters that were the same for all genetic algorithm schemes.

| Parameter                                | Value  |
|------------------------------------------|--------|
| Number of computations allowed           | 600    |
| Population size (without reduction)      | 30     |
| Number of generations (without reduction)| 20     |
| Number of crossover points               | 2      |
| Mutation rate                            | 0.05   |

### Table 3 | The frequency of the correct model being selected over 200 trials. The first 4 schemes are with the modifications and the last is without.

| GA scheme                          | Frequency of correct solution |
|------------------------------------|------------------------------|
| Adaptive $\Delta_{\text{ext}} = 0$ | 0.915                        |
| Adaptive $\Delta_{\text{ext}} = 0.1$ | 0.935                        |
| Adaptive $\Delta_{\text{ext}} = 0.2$ | 0.93                        |
| Adaptive $\Delta_{\text{ext}} = 0.3$ | 0.905                        |
| Bozdogan’s                         | 0.09                         |

### REFERENCES

Akaike, H. (1973). “Information theory and an extension of the maximum likelihood principle,” in Second International Symposium on Information Theory, eds B. N. Petrov and F. Csáki (Budapest: Académiai Kiadó), 267–281.

Boyce, D. E., Farhi, A., and Weischedel, R. (1974). Optimal Subset Selection: Multiple Regression, Interdependence, and Optimal Network Algorithms. New York: Springer-Verlag.

Bozdogan, H. (1988). “ICOMP: a new model-selection criterion,” in Classification and Related Methods of Data Analysis, ed. H. H. Bock (Amsterdam: Elsevier Science Publishers), 599–608.

Bozdogan, H. (2004). Statistical Data Mining and Knowledge Discovery. Boca Raton, FL: Chapman and Hall/CRC.

Fisher, R. A. (1930). The Genetical Theory of Natural Selection. Oxford: Clarendon Press.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

Hocking, R. R. (1976). The analysis and selection variables in linear regression. Biometrics 32, 1044.

Hocking, R. R. (1983). Developments in linear regression methodology: 1959–1982. Technometrics 25, 219–230.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press.

Mantel, N. (1970). Why stepdown procedures in variables selection. Technometrics 12, 591–612.

Moses, L. E. (1986). Think and Explain with Statistics. Reading, MA: Addison-Wesley.

Wilkinson, L. (1989). SYSTAT: The System for Statistics, Evanston, IL: SYSTAT.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 03 April 2010; paper pending published: 05 May 2010; accepted: 18 May 2010; published online: 08 July 2010.

Citation: Akman O and Hallam JW (2010) Natural selection at work: an accelerated evolutionary computing approach to predictive model selection. Front. Neurosci. 4:33. doi: 10.3389/fnins.2010.00033

This article was submitted to Frontiers in Systems Biology, a specialty of Frontiers in Neuroscience.

Copyright © 2010 Akman and Hallam. This is an open-access article subject to an exclusive license agreement between the authors and the Frontiers Research Foundation, which permits unrestricted use, distribution, and reproduction in any medium, provided the original authors and source are credited.