Summer School in Statistics for Astronomers & Physicists
June 15-17, 2005

Session on 'Computational Algorithms for Astrostatistics'

Genetic Algorithms

Max Buot
Department of Statistics
Carnegie-Mellon University

and

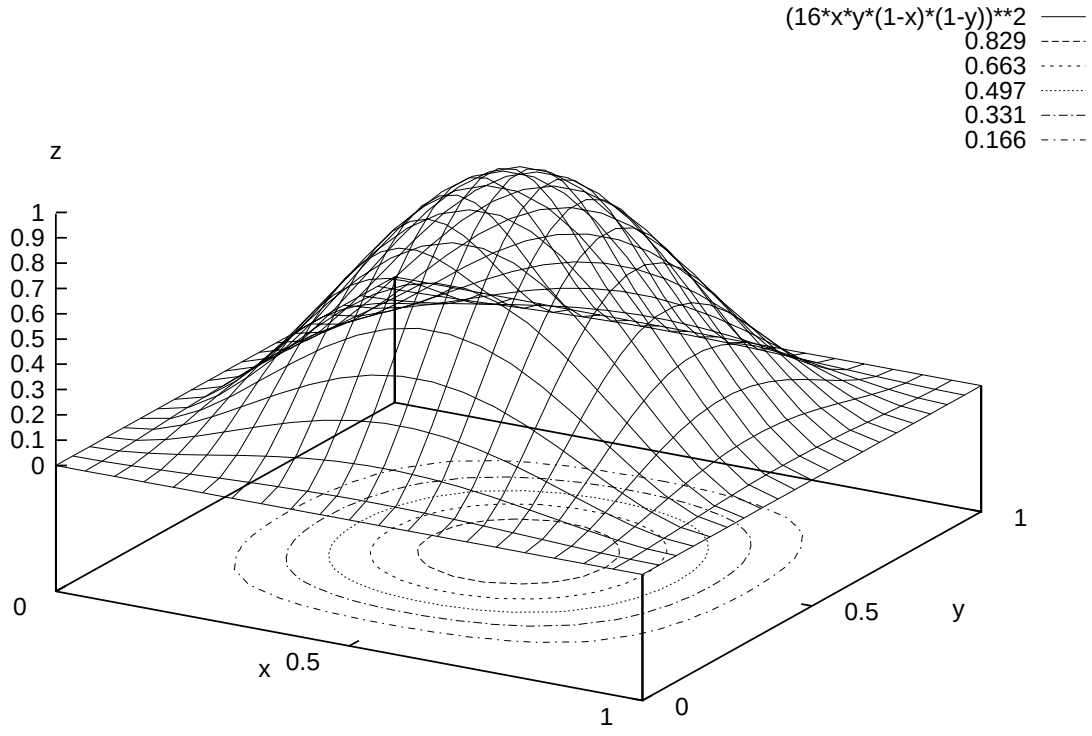
Donald Richards
Department of Statistics
Center for Astrostatistics
Penn State University

Genotype, phenotype, crossover, and mutation are ideas common to genetics. Genetic algorithms are computational algorithms incorporating these ideas from evolutionary biology.

Since their appearance in the 1960's, genetic algorithms are now widely used in astronomy and astrophysics, economics, finance, computer science, and other fields.

Easy problem: Find the point which maximizes

$$f(x,y) = [16\,x(1-x)y(1-y)]^2, \quad x,y \in [0,1]$$



Solution: $\widehat{x} = \widehat{y} = 1/2$.

The landscape is symmetric about $(\frac{1}{2}, \frac{1}{2})$.

Calculus: Solve $\frac{\partial f}{\partial x} = 0$, $\frac{\partial f}{\partial y} = 0$.

The method of steepest ascent:

Choose an initial guess (x_0, y_0).

Calculate the local gradient $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$.

Take a small step from (x_0, y_0), in the direction of maximum slope, to (x_1, y_1).

Calculate the gradient at (x_1, y_1).

Repeat the process until the maximum is attained.

Hill-climbing methods:
Steepest ascent method
Conjugate gradient method

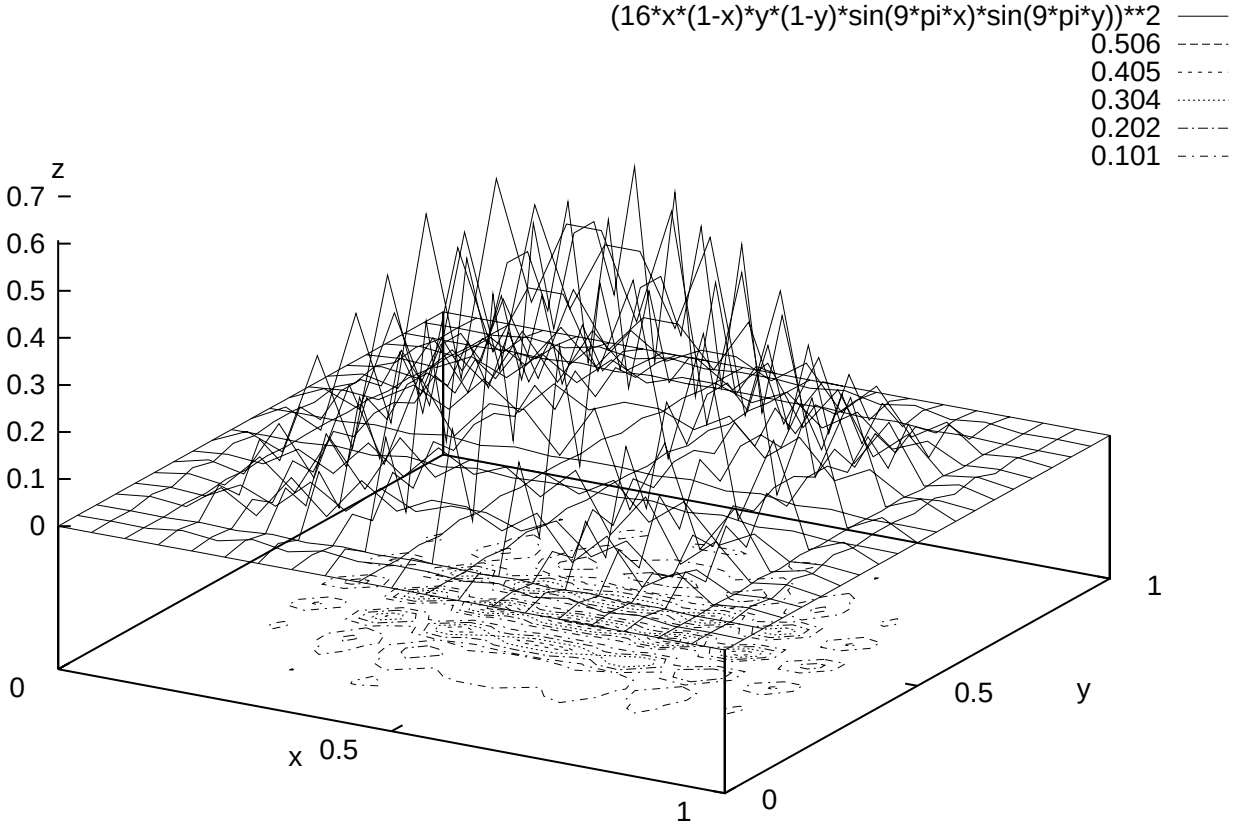
These methods work well if f is nice, smooth, and has only one global maximum.

New Example: Let

$f(x,y) = [16x(1-x)y(1-y)\sin(n\pi x)\sin(n\pi y)]^2$,

$0 \le x, y \le 1$, n is a positive integer.

For $n = 9$, f has 81 local maxima



Many of these maxima have nearly the same "height" and are separated by deep "valleys"

Hill-climbing works here if (x_0, y_0) is sufficiently close to the global maximum. Otherwise, hill-climbing converges to a local maximum.

For $n = 9$, steepest ascent locates the global maximum only if $\frac{4}{9} \leq x_0, y_0 \leq \frac{5}{9}$.

Charbonneau, "Genetic algorithms in astronomy and astrophysics," Ap. J. Suppl. Ser., 101:309-334, 1995 Dec.

Hill-climbing methods are:

Global: Can be applied to many types of functions, domains

Not Efficient: Often behave poorly in high dimensions or complicated domains

Not Robust: Small changes in the search space can cause them to behave erratically

We want optimization methods which are global, efficient, and robust

Simulated annealing methods

See *Numerical Recipes*

We also want optimization methods which:

Work with continuous or discrete variables

Require no information about derivatives

Search easily from vast portions of the domain

Can escape from local maxima

Can locate multiple global maxima

Can work with numerical data or analytical functions

Can work with encoded parameters

This is a lot to ask of any method.

Biological Evolution: An Optimization Process

Charles Darwin, *Origin of Species*, 1859

1831–1836: Darwin was the "naturalist" on HMS Beagle on its voyage around the world.

On return to England Darwin wrote, but did not publish, an essay on his theory of evolution.

1858: Alfred Wallace developed the same theory and published his results.

1859: Darwin published his theory.

Natural Selection: The process whereby individuals better adapted to their environment tend to produce more offspring, *on average*, than their less well-adapted competitors.

Darwin saw that two important ingredients were needed for natural selection to lead to large-scale evolution.

Inheritance: Parents will pass on their "fitness" to their offspring.

Variation: There exists a range of fitness levels over a population, and this allows natural selection to operate.

Gregor Mendel (late 1800's) later explained how inheritance is mediated and variation is maintained.

Francis Galton: Regression to the mean

R. A. Fisher found strong statistical evidence that Mendel's data were too good to be true.

Genotype: The genetic makeup of an individual, stored on "chromosomes" in the form of linear gene sequences.

Genes are carried on one of a pair of chromosomes in the form of DNA (**d**eoxyribo**n**ucleic **a**cid)

The DNA is in the shape of a double helix (Watson and Crick)

Each cell of the organism contains the same number of chromosomes

mosquitoes: 6 chromosomes per body cell

frogs: 26

goldfish: 94

humans: 46

Genotype: The genetic makeup of an individual, stored on chromosomes, in the form of linear gene sequences.

Phenotype: The actual individual which feeds, competes, and reproduces in the real-world environment.

"Genotype" refers to the genes present in an individual.

"Phenotype" refers to the observable traits or characteristics of an individual.

Development of "genetic algorithms."

Easy problem: Solve the equation $x^2 = 64$, where x is a positive integer.

Solution using a genetic algorithm (GA):

1. Construct a random initial "population" (we use binary arithmetic to construct a population of size four):

Binary string		Decimal
00100	$\longrightarrow$	4
10101		21
01010		10
11000		24

Each binary string is identified with a chromosome or individual.

2. Calculate each chromosome's fitness, e.g., we use the fitness function

$$\text{fit}(x) = 1{,}000 - |x^2 - 64|$$

The closer x^2 to 64, the higher fit(x).

Chromosome	Decimal	Fitness
00100	4	952
10101	21	623
01010	10	964
11000	24	488

3. Select individuals to become the parents of the next generation.

There are many ways to do this.

Delete the least fit chromosome and replace it with a copy of the most fit chromosome.

Parent pool: 00100 10101 01010 01010

4. Create the second generation from the parent pool. We will use *single-point crossover*:

Choose a pair of parent chromosomes at random, e.g., 00100, 01010

Cut the parent chromosomes at a randomly chosen place, e.g.,

$$\begin{array}{l|l} 001 & 00 \\ 010 & 10 \end{array}$$

Choose a crossover rate, p_c, $0 < p_c < 1$.

Generate a uniformly distributed random number R on $[0, 1]$. If $R \leq p_c$ then interchange the tails of the parent chromosomes:

$$\begin{array}{l|l} 001 & 00 \\ 010 & 10 \end{array} \longrightarrow \begin{array}{l} 01000 \\ 00110 \end{array}$$

This produces two children.

Repeat the process until we have four children.

Mutation: Choose a mutation rate p_m, $0 < p_m \leq 1$.

For each gene in each offspring, generate a uniform random number R on $[0, 1]$. If $R \leq p_m$ then mutate the gene from 0 to 1 or 1 to 0, as necessary.

Offspring pool		New generation
01 0 00		01 0 00
00 1 10	$\longrightarrow$	00 1 10
10 **1** 00		10 **0** 00
01 0 11		01 0 11

Replace the old generation with the new.

Return to Step 2: Calculate the fitnesses of the new generation.

Repeat the process until fitness converges to a maximum value of $1,000$.

Cerf (1996), p. 789

The selection, mutation, and crossover operators play different roles

Selection tends to concentrate the population on the best individuals currently existing

Mutation tends to disperse the population over the search space

Crossover tends to spread the information quickly over the population

If GAs in binary arithmetic seem to be time-consuming then bear in mind that DNA is coded in quaternary arithmetic (base four).

DNA consists of four types of nucleotides which differ in only one component, a base which contains nitrogen.

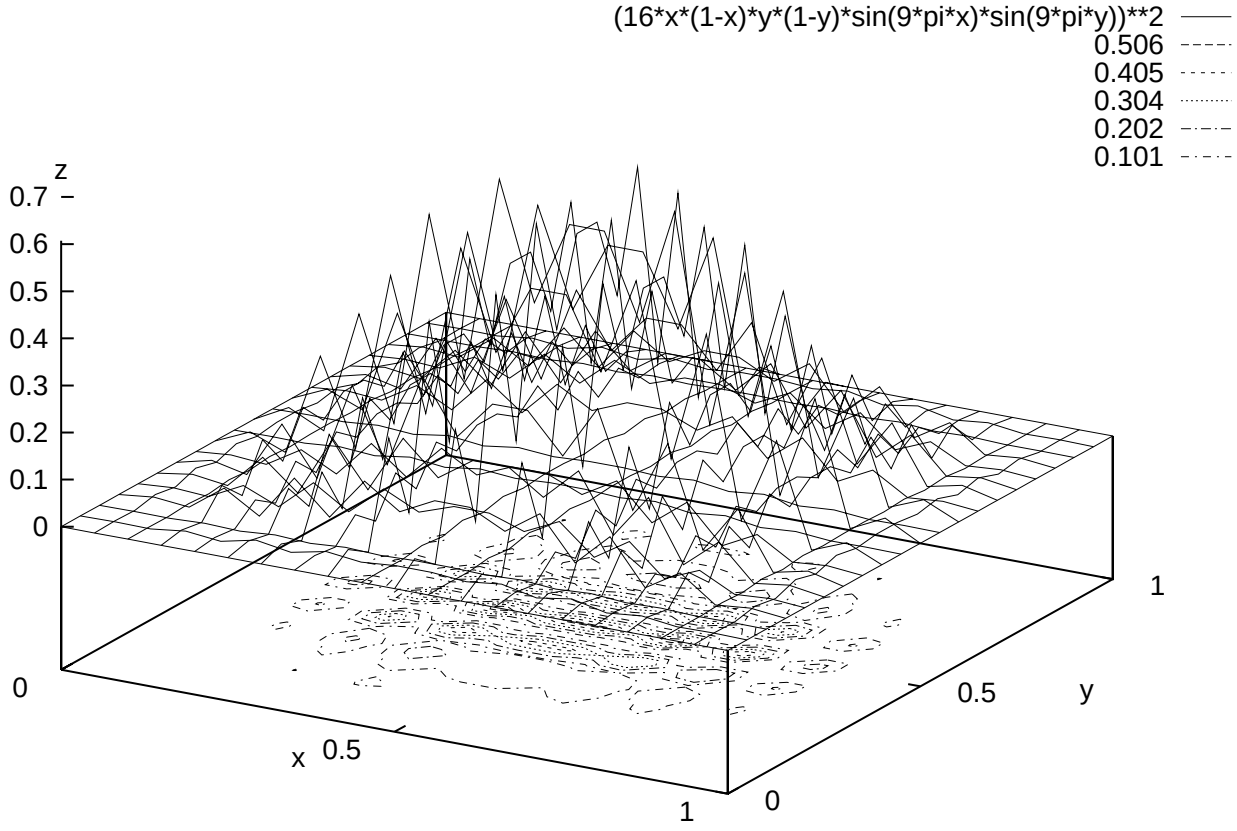
The bases are: **A**denine, **C**ytosine, **G**uanine, and **T**hymine.

The movie *Gataca*

An example in decimal arithmetic

$f(x,y) = [16x(1-x)y(1-y)\sin(n\pi x)\sin(n\pi y)]^2$,

$0 \le x, y \le 1$, n is a positive integer.



Variations on Genetic Algorithms

Choose at random an initial population

Compute the fitness of each chromosome

Thresholding: Choose a cutoff fitness level, f_{min}. Delete chromosomes with fitness below f_{min}.

Pairing: How do we choose pairs of parents?

Choose pairs of parents at random so that all pairs have an equal chance of being selected?

Rank chromosomes by fitness, $f_1 \geq f_2 \geq \cdots$. Pair chr_1 with chr_2, chr_3 with chr_4, etc.

These are not good models of Nature.

Weighted pairing: Use the fitness levels of the mating pool to form a probability distribution,

$$\pi_i = \frac{f_i}{f_1 + \cdots + f_n}, \quad i = 1, \ldots, n.$$

Choose parent pairs at random according to this empirical distribution of fitness levels.

The chance that a chromosome is selected to be a parent is proportional to its fitness level.

Rank weighting: Use the ranks of the mating pool to form a probability distribution. Choose parents at random using this distribution.

Single-point crossover

Two-point crossover with two parents

$$\begin{array}{ll} \text{parent 1} & \underbrace{10}_{A} \,\|\, \underbrace{11}_{B} \,\|\, \underbrace{101}_{C} \\ \text{parent 2} & \underbrace{01}_{A} \,\|\, \underbrace{01}_{B} \,\|\, \underbrace{100}_{C} \end{array}$$

Select one of A, B, or C, and interchange the corresponding genes. If, say, B is chosen then the resulting offspring are

$$\begin{array}{ll} \text{offspring 1} & 10 \,|\, \mathbf{01} \,|\, 101 \\ \text{offspring 2} & 01 \,|\, \mathbf{11} \,|\, 100 \end{array}$$

Two-point crossover with multiple parents

parent 1	010**10101**0101
parent 2	1111**1110**0000
parent 3	1100**1100**1001
offspring 1	010**10101**0000
offspring 2	010**10101**0011
offspring 3	1111**1110**0101
⋮	⋮
offspring 18	1111**0101**0011

Elitism: A copy of the fittest chromosome in each generation is included in the next generation.

Variable crossover or mutation rates: Decrease these rates as the algorithm nears convergence.

Choosing the new generation:

Full generational replacement: Accumulate offspring in temporary storage; when enough have been created then they replace the entire parent population.

Steady state reproduction: Add offspring to the parent pool as soon as they are bred.

Hybrid methods: Use GAs to obtain good estimates of the location of the global maximum, then apply hill-climbing methods.

Drawbacks of Genetic Algorithms

GAs are not guaranteed to find the maxima

GAs locate maxima almost surely

R. Cerf (1998), *Adv. Appl. Probab.*

Rigorous analysis of convergence of GAs

A brief summary of Cerf's results

E : The set of all binary strings of length N

f : A positive fitness function defined on E

f^* : The set of *all* global maxima of f on E

Our GA consists of:

$\cdots \rightarrow$ mutation $\rightarrow$ crossover $\rightarrow$ selection $\rightarrow \cdots$

Selection: Choose parents independently and at random from current candidates using the following probability distribution: If the lth generation is $\{v_1, \ldots, v_m\}$ then the $(l+1)$th generation is $\{x_1, \ldots, x_m\}$ with probability

$$\prod_{k=1}^{m} \left(\frac{l^{cf(x_k)}}{\sum_{k=1}^{m} l^{cf(v_k)}} \right)^{x_k(i)}$$

Hamming distance: Given two chromosomes $i = (i_1, \ldots, i_N)$ and $j = (j_1, \ldots, j_N)$, the Hamming distance between i and j is the number of letters where i and j differ; e.g.,

$$H(100110, 010101) = 1+1+0+0+1+1 = 4$$

Mutation: At the lth generation, the probability that chromosome i is mutated into chromosome j is

$$p_l(i,j) = \begin{cases} 0, & \text{if } H(i,j) > 1 \\ l^{-a}, & \text{if } H(i,j) = 1 \\ 1 - Nl^{-a}, & \text{if } H(i,j) = 0 \end{cases}$$

where $a > 0$ is a constant

At the lth generation, the probability of mutating the population $\{x_1, \ldots, x_m\}$ to obtain $\{u_1, \ldots, u_m\}$ is the product

$$p_l(x_1, u_1) \cdots p_l(x_m, u_m)$$

For large l, very little mutation takes place

Crossover: At the lth generation, the probability that a pair of chromosomes (i, j) is "crossed-over" into a pair (i', j') is $q_l((i, j), (i', j'))$. The function q is defined as follows:

1. Define "cutting" functions $T_1, \ldots, T_{N-1}$ from $E \times E$ onto $E \times E$. Let $i = i_1 \cdots i_N$ and $j = j_1 \cdots j_N$. Then T_k maps (i, j) to (i', j') where

$$i' = i_1 \cdots i_k j_{k+1} \cdots j_N, \quad j' = j_1 \cdots j_k i_{k+1} \cdots i_N$$

Pictorially,

$$T_k : \begin{array}{c} i_1 \cdots i_k i_{k+1} \cdots i_N \\ j_1 \cdots j_k j_{k+1} \cdots j_N \end{array} \rightarrow \begin{array}{c} i_1 \cdots i_k j_{k+1} \cdots j_N \\ j_1 \cdots j_k i_{k+1} \cdots i_N \end{array}$$

2. Define the function

$$\beta((i, j), (i', j')) = \#\{k : 1 \leq k \leq N-1, T_k(i, j) = (i', j')\}$$

$\beta((i, j), (i', j'))$ is the number of cutting functions which transform (i, j) into (i', j')

3. For any two pairs (i, j) and (i', j'), define

$$q_l((i, j), (i', j')) = \beta((i, j), (i', j'))l^{-b}$$

whenever $(i, j) \neq (i', j')$, and

$$q_l((i, j), (i, j)) = 1 - \sum_{(i',j') \neq (i,j)} \beta((i, j), (i', j'))l^{-b}$$

where $b > 0$ is a constant

At the lth generation, the probability that crossover applied to (i, j) results in (i', j') is $q_l((i, j), (i', j'))$

Very little crossover takes place when l is large

m : The population size

N : The number of digits in each binary string

a, b, c : Constants controlling the mutation rate, crossover rate, and selection probabilities, resp.

f^* : The set of *all* global maxima of f on E

$\delta = \min\{|f(i) - f(j)| : i, j \in E, f(i) \neq f(j)\}$

$\Delta = \max\{|f(i) - f(j)| : i, j \in E, f(i) \neq f(j)\}$

Theorem (Cerf, 1998): Suppose that

$$m > \frac{aN + c(N-1)\Delta}{\min\{a, b/2, c\delta\}}.$$

Then, regardless of the choice of initial population, as $l \to \infty$,

$$P(\text{The entire } l\text{th popn. is in } f^*) \to 1.$$

Cautionary Remarks

To calculate values of m needed for convergence with probability 1, we must *know* the values of δ and Δ

To evaluate δ and Δ, we need to know the values of f at its global maxima and minima

Paradoxical problem: Knowing the maximum value of f at its global maxima makes it easier to locate the maxima

Convergence times for GAs: epochs, cutoff phenomena,

Comparison with the EM and Monte Carlo algorithms