

evolMC: a package for Monte-Carlo simulation

W Burchett E Roualdes G Weyenberg

Fall 2013

Getting the package

Package evolMC.

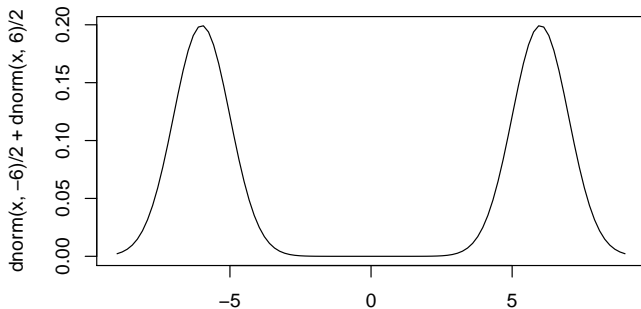
<http://github.com/grady/evol-mc>

```
library(devtools)
install_github("evol-mc", "grady")
library(evolMC)
```

See the “example” package vignette for demo code.

The Problem

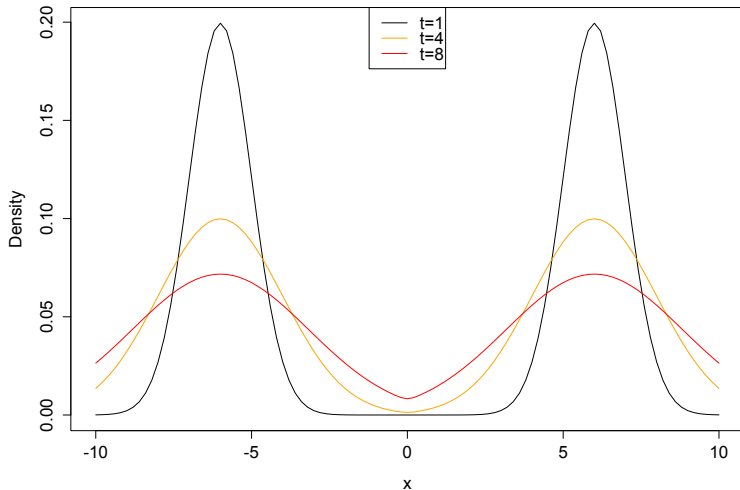
- ▶ We wish to sample from a distribution with multiple modes, separated by regions of very low probability density.
- ▶ Traditional Metropolis algorithms do not work well here.



Heating a Distribution

- ▶ Consider a distribution $f(x|t) \propto \exp\{-H(x)/t\}$, where $t \geq 1$ is called the *temperature* and $H(x)$, called the *fitness function*, corresponds to the negative log-density of x , up to a constant.
- ▶ Increasing the temperature causes the distribution to become “flatter”, i.e. to have thicker tails.

Heating a Distribution



Genetic Algorithm Language

- ▶ A *population* \mathbf{x} consists of N individuals x_i , $i = 1, \dots, N$, where $x_i \in \mathbb{R}^d$.
- ▶ Associated with the population is a set of temperatures $\mathbf{t} = (t_1, \dots, t_N)$. It is assumed that the t_i are in descending order, terminating at the temperature of the target distribution. (In most cases this means $t_N = 1$.)
- ▶ Each individual x_i is independently sampled from the distribution $f_i(x_i) \propto f(x_i|t_i)$.

Algorithm 1 Evolutionary Monte Carlo

procedure EMC

with prob p_m

 MUTATE

otherwise

 CROSSOVER

end w/prob

 EXCHANGE

end procedure

▷ p_m is the *mutation probability*.

Algorithm 2 A Gaussian random-walk *mutation*.

procedure MUTATE

Copy the current population to x .

for all individuals x_i in x **do**

$y \leftarrow \mathcal{N}_d(x_i, t_i \sigma^2 I)$

with prob $\min\{1, \exp(-H(y)/t_i + H(x_i)/t_i)\}$

$x_i \leftarrow y$

end w/prob

end for

Set current population to x .

end procedure

Algorithm 3 The fitness-weighted *crossover*.

procedure CROSSOVER

Copy the current population to x .

for all individuals x_i in x **do**

$w_i \leftarrow \exp(-H(x_i))$

end for

Select k uniformly from $\{1: d\}$.

Select i from $\{1: N\}$ with weights proportional to $\{w.\}$.

Select j uniformly from $\{1: N\} \setminus \{i\}$.

In x , swap elements $k: d$ of individuals i and j .

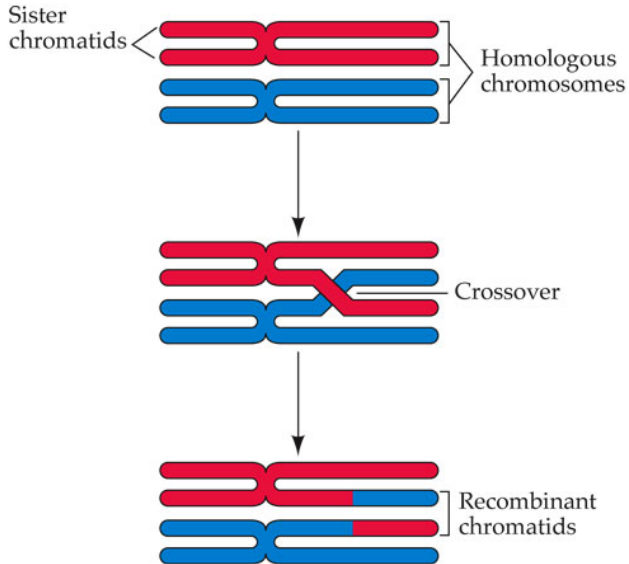
with prob \dots

▷ Metropolis update.

Set the current population to x .

end w/prob

end procedure



LIFE: THE SCIENCE OF BIOLOGY, Seventh Edition, Figure 9.16 Crossing Over Forms Genetically Diverse Chromosomes
© 2004 Sinauer Associates, Inc. and W. H. Freeman & Co.



Algorithm 4 The *exchange* attempts to swap individuals between neighboring temperature states.

procedure EXCHANGE

Copy the current population to x .

Select i uniformly from $\{1: N\}$.

Select j uniformly from $\{i \pm 1\} \cap \{1: N\}$.

Swap individuals i, j of x .

with prob $\min\{1, \exp(-H(x_i)/t_i - H(x_j)/t_j + H(x_i)/t_j + H(x_j)/t_i)\}$

Set the current population to x .

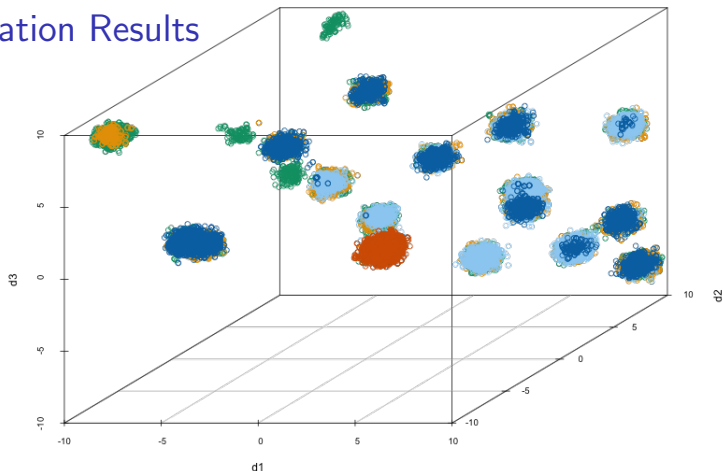
end w/prob

end procedure

Simulation Setup

- ▶ Target distribution: 20 part $\mathcal{N}(\mu, 0.1I_3)$ w/ equal mixing proportions and $\mu \in [-10, 10]^3$.
- ▶ Mutation probabilities: $p_m \in \{0.1, 0.5, 0.9, 1.0\}$
- ▶ Mutation updater: $u_m \in \{\text{uniform}, \text{Gaussian}\}$
- ▶ Temperatures: $\mathbf{t} \in \{(1, 5), (1, 2, 4, 8, 15), (1, 5, 15, 30, 50)\}$
- ▶ MH: multivariate normal proposal distribution
- ▶ Iterations: $2e4$

Simulation Results



coverage of modes by updater

