Notes for Cajo J. F. Ter Braak "A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces"

Malte Brinch
University of Copenhagen
Septemper 2019

1. Introduction

Differential Evolution (DE) is a simple genetic algorithm for numerical optimization in real parameter spaces. In a statistical context one would not just want the optimum but also its uncertainty. The uncertainty distribution can be obtained by a Bayesian analysis (after specifying prior and likelihood) using Markov Chain Monte Carlo (MCMC) simulation. Differential Evolution Markov Chain (DE-MC)

The idea is to run multiple Markov chains in parallel and let them learn from each other. The combination of DE and MCMC solves an important problem in MCMC in real parameter spaces, namely that of choosing an appropriate scale and orientation for the jumping distribution. Normally the proposal (jumping) distibution is a multivariate normal distribution. There are problems with specifying the covariance matrix for the distribution. The variances and covariances need to chosen so as to balance progress in each step and a reasonable acceptance rate. If parameters are highly correlated, special precautions must be taken to avoid singularity of the estimated covariances matrix.

The idea is to run N chains are run in parallel and the jumps for a current chain are derived from the remaining N-1 chains. The simplest way to do this for a single jump is to use two other randomly selected chains multiply them by a factor γ and then add that vector to the current chain. This gives information about both the scale and orientation of the jump.

2. Theory

2.1. Random walk metropolis

RWM draws a sample from a d-dimensional target distribution with probability density function. it updates a single parameter at a time by generating a proposal, calculating its metropolis ratio (how likely is it compared to the old value) and accepts it with a given probability or rejects it and keeps the old value. The proposal is $x_p = x + \epsilon$ where ϵ $N(0, \Sigma^{\sim})$. The variance matrix is chosen by the user and optimally has $\Sigma^{\sim} = c^2\Sigma$ with Σ=covariance(x) and $c = 2.38/\sqrt{d}$ where d is the number of dimensions of the parameter vector x

2.2. Genetic algorithms and differential evolution

Here markov chains are done in parallel, where the state of a chain is given by a parameter vector x with dimension d but now there are multiple chains and the vectors are part of a population X a N times d matrix. Differential evolution is done through schemes like the following

$$x_p = x_{R0} + \gamma(x_{R1} - x_{R2})$$

where x_{R0}, x_{R1} and x_{R2} are randomly selected without replacement from the population X_{-i} (the population without x_i). The proposal vector is retained if the fitness of x_p is higher than the fitness of x_i. to introduce this idea into a Markov chain the proposal and acceptance scheme must be such that there is detailed balance with respect to the probability density function. The scheme is modified such that

$$x_p = x_i + \gamma(x_{R1} - x_{R2}) + e$$

where e is drawn from a symmetric distribution with a small variance compared to that of the target, but with unbounded support, e.g. $e \sim N(0, b)^d$ with b small. For large N and small b we will have

$$x_p = x_i + \gamma\epsilon$$

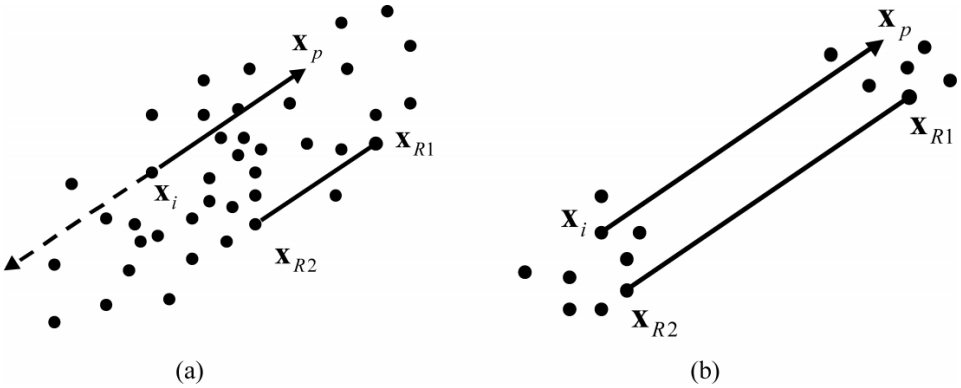

Fig. 1 Differential Evolution in two dimensions with 40 (a) and 15 (b) members in the population ($d = 2$, $N = 40$ and 15). The proposal vector $\mathbf{x}_p$ to update the ith member is generated from $\mathbf{x}_i$ and the randomly drawn members $\mathbf{x}_{R1}$ and $\mathbf{x}_{R2}$ by (2) with $\gamma = 2.4/(2 \times 2)1/2 = 1.2$ in (a) and $\gamma = 1.0$ in (b) and $\mathbf{e} = (0, 0)$ in both. The dashed arrow in (a) points to the proposal when $\mathbf{x}_{R1}$ would have been drawn after $\mathbf{x}_{R2}$. The reverse jump from $\mathbf{x}_p$ to $\mathbf{x}_i$ is obtained by translating the dashed arrow to $\mathbf{x}_p$.

where E(ϵ)=0 and cov(ϵ)=2Σ which is like RWM, so we have $\gamma = 2.38/\sqrt{2d}$ as a optimal choice. the value of γ changes the acceptance probability from 0.44 for d=1 to 0.23 for d large.

There is a difference between Gibbs and DE-MC sampling. The proposals in Gibbs sampling are drawn from the appropriate conditional distribution. The proposals in DE-MC for a particular dimension are generated, after convergence, from differences of two numbers drawn from the marginal distribution for that dimension. This shows that crossover in DE-MC would work best (as in Gibbs) if the dimensions that are updated in separate steps are independent.

If blocks are strongly correlated, γ may need to be decreased.

The number of parallel chains can go from 2d to 3d for simple function