Project : Monte Carlo computation of statistical moments

Vincent Pollet
vincent.pollet@epfl.ch

Clément Lefebvre
clement.lefebvre@epfl.ch

December 16, 2016

1 Features

This program computes the statistical moments of a function, using a Monte Carlo method to compute the integrals. The following features are available :

- Compute any moment of unary discretely defined functions (ie functions defined by a set of data points $(x, f(x))$, stored in a CSV file).
- Compute any moment of Continuously defined multi-dimensional functions (implemented in a lua source file), only in square, cubic, hypercubic ... domains.
- Random samplings used in computation can follow three different distributions : Uniform, Normal, Exponential.
- Interpolation for discrete functions with four different methods : Linear, Closest, Lower, Upper.

2 Compiling the program

To compile the program the following step must be followed :

- If Lua is installed on the computer
  - Get in the /pcsc_montercarlo directory in the terminal.
  - Generate the Makefile with CMake command : "ccmake ."
  - Compile the main program with "make MCMoment_Calculator"
  - Compile the test suite with "make Test_Suite"
  - The program binary is inside the "/bin" folder and the test suite inside the "/Test_Suite" folder.
- If Lua is not installed on the computer
  - Get in the /pcsc_montercalo/lua directory in the terminal.
  - Run the command "make xxx" where xxx is your platform : aix bsd c89 freebsd generic linux macosx mingw posix solaris
  - Run the command "make install"
  - Then do the same steps than before to compile the program.

  See the known bugs section if you encounter problems in these steps

3 Configuration

The program requires a configuration file to start. The configuration file must be written in the following format : One argument per line, which value is entered between " specifiers (cf example in listing 1). All arguments must be written in the given order, even those not used in the simulation. For instance an interpolation method must be specified even if the function is continuously defined, and thus will not be interpolated.

The arguments and their possible values are the following :

Type of Function : It tells the program how the function, which moments are to be computed, is defined. This parameter can take only 2 values :

- ***Continuous*** The function is actually implemented in a lua file.
- ***Discrete*** The function is defined by a set of pairs $(x, f(x))$ in a CSV format. The function will be interpolated between the specified values, with the interpolation method prescribed by the user (cf interpolation method below)

Continuous functions can be defined from $\mathbb{R}^n$ to $\mathbb{R}$, with n an integer. Discrete function can only be defined from $\mathbb{R}$ to $\mathbb{R}$.

Dimension : The dimension of the arguments of the function. If f goes from $\mathbb{R}^d$ to $\mathbb{R}$, the dimension is d. Discrete functions are only accepted with $d = 1$!

Interpolation method : The method used to interpolate a *discrete* function between the data points. Available methods are :

- ***Linear*** The function is approximated by linear polynomial between data points
- ***Upper*** The function is approximated in x by taking the data closest to x, but superior to x.
- ***Lower*** The function is approximated in x by taking the data closest to x, but inferior to x.
- ***Closest*** The function is approximated in x by taking the data closest to x, superior or inferior.

Function file path : The path to the file where the function is defined. It can either be the absolute path, or the relative path from the executable of the program.

Random distribution : The random distribution of the evaluation points for the Monte Carlo integration used when computing the moments. Available distributions are :

- ***Uniform*** The points are uniformly distributed over a specified interval. The Interval boundaries are specified in the ***Domain inferior boundary*** and ***Domain superior boundary*** fields.
- ***Normal*** The points distribution is Gaussian. The mean of the distribution must be specified in the ***Normal distribution mean*** field and the standard deviation in the ***Normal distribution standard deviation*** field.
- ***Exponential*** The points distribution is exponential. The parameter of the distribution (λ) must be specified in the ***Exponential distribution lambda parameter*** field.

Random Generator seed : The seed can be used to generate the same sequence of number multiple times. With the value 0, the seed will be chosen based on the cpu clock at the time the generator is instantiated in the program.

Number of evaluation points : The number of points that will be randomly picked to compute the moments.

Number of simulations : The number of times the moment will be computed.

```
Type of Function                                 "Discrete"
Dimension                                        "1"
Interpolation method                             "Linear"
Function file path                               "FunctionDataPoints.dat"
Random distribution                              "Uniform"
Domain inferior boundary                         "0"
Domain superior boundary                         "1"
Normal distribution mean                         "0.1"
Normal distribution standard deviation           "1"
Exponential distribution lambda parameter        "0.33"
Random generator seed                            "0"
Number of evaluation points                      "1000"
Number of simulations                            "10000"
```

Listing 1: Example of configuration file

4 Program execution

The following program typical execution follows the following steps :

- A MomentComputerManager object is declared. It reads a configuration file previously constructed by the user. An ExceptionManager must also be created to handle the eventual errors.
- A RandomGenerator, Function and MCSimulator are created depending on the configuration of the user.
- Finally the moments can be computed by the user using the function ComputeMoments().
- Moments are written to a file specified by user. The moments distribution can be visualized with the provided Matlab script "Plot_Moments_Distribution.m". An example of an output file : "Moment_order_1_f_x_equal_x_square_uniform_between_0_and_1" is available in the "README" folder.

5 Perspective & Limitations

- Improve the configuration file reading : allow to write only useful parameters for the simulation, allow the use of fraction number syntax (e.g $\frac{1}{3}$ instead of 0.3333..)
- Implement multi-dimension computations for discrete functions and allow multi-dimension domain definition.
- Add better interpolation methods for discrete functions.
- Implement error management for the Lua files.
- Implement a graphical library to plot the moments distribution.

6 List of tests

To run the different tests, you only have to run the binary "Test_Suite" located in the "Test_Suite" folder. Take care to not delete the "fx=x2", "Continuous_Function_Test1.lua" and the "Continuous_Function_Test2.lua" files.

The tests use the program to compute moments of different orders, using the function defined in the previously mentioned files. The values are then compared to the analytical results : the test is considered passed if the relative error on the computed values is less than 1 %. Since the Monte Carlo points are chosen randomly, the results follow a normal distribution (conclusion of the central limit theorem) around the analytical value. To minimize the random aspect, the value over 100 simulations are averaged in the tests.

Test 1 First and second moments of a continuous function $f(x) = x$, uniform distribution between $[0, 1]$.

Test 2 First and second moments of a continuous function $f(x) = x$, normal distribution of parameter $\mu = 2$, $\sigma = 3$.

Test 3 First moment of a continuous function $f(x) = x$, exponential distribution with $\lambda = \frac{1}{3}$.

Test 4 First moment of a discrete function $f(x) = x^2$, uniform distribution between $[0, 1]$.

Test 5 First and second moments of a continuous function $f(x, y) = x+y$, uniform distribution between $[2, 3]$.

Test 6 First moment of a continuous function $f(x, y) = x+y$, normal distribution of parameter $\mu = 2$, $\sigma = 3$.

Test 7 First moment of a discrete function $f(x) = x^2$, uniform distribution between $[1, 3]$.

Final test Compare the computed moments of "Test 7" with the output file to be sure that they are identical.

7 Known bugs

7.1 Compilation

- There is a possibility that the lua library can't be build if the readline library is missing. One solution is to run the command line "apt-get install libreadline-gplv2-dev" in the terminal.
- When compiling the program, if there are some undefined reference (e.g 'dlsym', 'dlerror', 'dlopen') modify the line in the CMakeList :

```
    set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -std=c++11")

```

Listing 2: Original line

into

```
    set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -std=c++11 -Wl,--no-as-
  needed -ldl")

```

Listing 3: New line