PAMC C code

This is the C version of Precision Annealing Monte Carlo (PAMC) method. For more detail on the method, please read _____(paper not ready yet, will update later when it is ready).

Brief description of PAMC method

Assuming we have a system with model

$$x_a(n+1) = f_a(\vec{x}(n), \vec{p}) + \eta$$

Where x is the state variables, p is the list of parameters to be estimated, η is a noise term of the model. n is the index of the system. For example, n can be the index of layers in a neural network, or time step in a dynamical systems. Subscript a satisfies $0 \leq a \leq D$, and D is the number of total dimensions of the model.

Furthermore, we have a set of incomplete and noisy measurements as well: $\mathbf{Y} = \{\vec{y}(0), \vec{y}(1), \ldots, \vec{y}(n), \ldots, \vec{y}(N)\}$. Measurements $\mathbf{Y}$ is on L dimensions and $L \leq D$.

Assuming all noises are Gaussian, for each path $\mathbf{X} = \{\vec{x}(0), \vec{x}(1), \ldots, \vec{x}(n), \ldots, \vec{x}(N), \vec{p}\}$, the conditional probability distribution of $\mathbf{X}$ given measurements $\mathbf{Y}$ can be written as $P(\mathbf{X}\,|\,\mathbf{Y}) = \frac{1}{Z} exp(-A(\mathbf{X}, \mathbf{Y}))$, where Z is the normalization factor, $A(\mathbf{X}, \mathbf{Y})$ is the negative log likelihood and is often referred to as action or cost function. The action has form

$$A(\mathbf{X}, \mathbf{Y}) = \frac{1}{2} R_m \sum_{n=0}^{N} \sum_{l=1}^{L} (x_l(n) - y_l(n))^2 + \frac{1}{2} R_f \sum_{n=0}^{N-1} \sum_{a=1}^{D} (x_a(n+1) - f_a(\vec{x}(n)))^2$$

The first term in $A(\mathbf{X}, \mathbf{Y})$ is convex and quadratic, which is convenient since we can easily find the minimum of $A(\mathbf{X}, \mathbf{Y})$. However, the second term involves the non-linear model equation $f_a(\vec{x}(n))$, which makes $A(\mathbf{X}, \mathbf{Y})$ non-convex and contains many local minima. PAMC approach to this problem is to start with a small R_f values, and then slowly increase the R_f value as we track the minimum of $A(\mathbf{X}, \mathbf{Y})$. With each fixed R_f value, we do a Metropolis-

Hasting Monte Carlo sampling to find the expected value of variables and unknown parameters.

Overview of PAMC process

First, equations of the model, constants and parameters of the model needs to be written in a header file (I would usually name them as model_***.h, where *** refers to a specific model). Then the measurements, initial guesses of the variables and unknown parameters needs to be given, either in the specs.txt file as I explain later, or in .dat files, then providing the file path in specs.txt. specs.txt also sets up some other hyper parameters of the PAMC process, which will be explain in the following section.

To run the program, first modify the MakeFile, especially make sure that it uses the right model_***.h file. Then run with command like "./PAMC_main 0 ./Lorenz96". Here 0 is the number of trials we are running, since we may use a script to run a batch of trials. "./Lorenz96" is the path to model specific files (model_***.h, initial conditions, measurements, specs.txt, etc.).

The program will start with a small R_f values, do $Nit = Nini + Bsize * Nblock$ iterations. Here $Nini$ is the number of iterations during initialization phase. Nblock is the number of blocks during the main MC procedure, and Bsize is the size (number of iterations) of the block. Within each iteration, each variable and unknown parameter will perturb once by adding a random value from a uniform distribution $[-\delta x, \delta x]$, the accept the change with a probability $max(1, exp(-dA))$, where dA is the change in action due to the change in variable/parameter. δx is adjusted during the initialization phase, and is different for different variables. The adjustment is simply calculating the average acceptance rate r for a certain number of iterations in initialization phase, and update based on $\delta x \leftarrow \delta x(1 + a(r - r_{expected}))$.

After $Bsize$ iterations in the main procedure, the average of the block is calculated and at the end, the average of all the block is calculated as the expected value respect to this specific R_f. Then R_f is increased as $R_f \leftarrow R_f * \alpha$. The process above is repeated. The program stop until $R_f = R_{f0}\alpha^{\beta_{max}}$.

Details of the inputs and outputs

model_***.h

There are a few things have to be define here:

1. D: Number of variables, or total dimension of variables
2. M: Total number of index n as in $x_a(n+1) = f_a(\vec{x}(n), \vec{p}) + \eta$
3. NP: Total number of unknown Variables
4. beta_max: Number of annealing steps as in $R_f = R_{f0}\alpha^{\beta_{max}}$
5. alpha: Size of the annealing step as in $R_f = R_{f0}\alpha^{\beta_{max}}$
6. N_ini: Number of iterations of initialization phase
7. Bsize: Block size for the main procedure
8. Nblock: Number of blocks for the main procedure
9. delta_0: Default δ_x for variables at the beginning
10. deltaF_0: Default δ_x for unknown parameters at the beginning
11. f(d,m,y,p): the equations for the model. d and m are the index of D and M dimension, y is the measurement, and p is the list of parameters to be estimated

specs.text

specs.txt contains several file paths and hyper parameters to specify. Each line starts with 2 capital letters in order for the main program to recognize and read it right. I will describe each line here.

"AP": Annealing parameters, namely R_m and R_{f0} to start with. This is used if R_m and R_f is the same for all the variables, for example, Lorenz 96 model.

"AF": DO NOT use together with "AP". Annealing parameters, but are specified in 2 files, for R_m and R_{f0} respectively. This is convenient if R_m and R_f are different for different variables. Inside the file, R_m and R_{f0} need to be specified for all dimensions.

"SS": δx adjustment during the initialization phase. First is a and second is $r_{expected}$.

"LF": the fraction/percentage of measured variables. This is used for a twin experiments or test case. Then the index of measurements will be assigned equally spaced on total dimensions.

"OD": DO NOT use together with "LF" File contains measured index. first number is the number of measured dimensions, and second is the path of the file. This is useful for the actual experiments.

"MP": Path for the measurement file. The first number tells how many variables are included in the file. Usually it is equal to D if we are doing twin experiments/test case, or L if we are running PAMC on the actual experiments. The second is the path for the measurement file.

"IP": Path for the initial guess on variables. Note that this file should be in the format of ***_n.dat, where n is the index of trials as we run the command "./PAMC_main n ./Lorenz96" for example. *** needs to be put here in this line.

"PA": Initial guess for the unknown parameter if there is only one.

"PF": File contains initial guesses for all the unknown parameters. This have not been implemented yet since I don't have the necessity at the moment, but will update later.

output files

The action after each annealing step, as well as the first (measurement error term) and second (model error term) term in the action is written in action.dat file, in the oder of action, measurement term and model term.

The parameter estimation after each annealing step is written in parameter.dat file.

The path estimation at the end of whole PAMC process is written in path.dat file.

Included examples

Lorenz 96 with $D = 20$

This is a simple example with 20 variables and 1 parameter. It is a test case for the program.

Shallow water model

This is a model in geophysics to describe the fluid when the depth of the fluid is much small compared to the horizontal size. This example is still under development. Hopefully it will be finished soon.