

# Introduction to the DeCo MATLAB Toolbox

January 9, 2015

## 1 License, Hardware and Software requirements for the DeCo Toolbox

Name of the Package: The DeCo MATLAB Toolbox.

Type: Density combination.

Release Date: 2014-05-23. This version 2014-07-07

Authors: Roberto Casarin, Stefano Grassi, Francesco Ravazzolo and Herman K. van Dijk.

Maintainer: Stefano Grassi <S.Grassi@kent.ac.uk>

License: GPL ( $\geq 2$ ).

Software requirements: Matlab 2012b or greater with parallel toolbox.

Hardware requirements: Standard PC with possibly NVIDIA Video Card (CUDA enabled) with computing capability 1.3 or greater, see <http://docs.nvidia.com/cuda/index.html>.

## 2 General description of the DeCo Toolbox

The **DeCo** toolbox is explicitly designed to run dataset of our examples or other user's dataset. The toolbox does not reproduce the Monte Carlo simulation exercises of the paper, more precisely Figure 1, Figure 2 and Table 2 of the paper. To replicate those results please refer to replication files directory.

The DeCo toolbox is composed by five directories and four files. Those are:

- CodeCPU, contains the code for the CPU program;
- CodeGPU, contains the code for the GPU program;
- Dataset, contains the Dataset used by the program;
- OutputCPU, Output of the CPU program, in .mat format;
- OutputGPU, Output of the GPU program, in .mat format;
- DeCo\_GUI.fig, main file for the GUI;

- DeCo\_GUI.m, main execution file for the GUI;
- DeCo\_CPU, function to run the CPU program without using the GUI;
- DeCo\_GPU, function to run the GPU program without using the GUI.

To execute the program using the GUI<sup>1</sup> just open the file DeCo\_GUI.m with the MATLAB editor and execute the script. A GUI similar to Figure 1 should be displayed if a supported GPU card is present, otherwise a GUI similar to Figure 2 will be displayed, where only CPU bottom is enabled and "Block of Draws" is not visible.

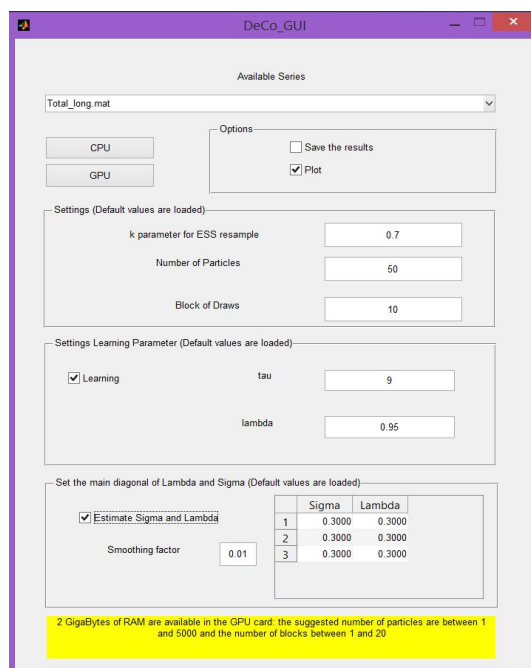


Figure 1: The graphical user interface of the **DeCo** package.

The GUI allows the user to set some key parameters (see also the related paper), those are:

- The *ListBox* loads and displays the available datasets in the directory Dataset.
- The *Panel* "Options" contains the command for saving and plotting (selected by default) the results. The results are saved in the directory *OutputCPU* or *OutputGPU* depending on the device used for the calculation, the CPU or the GPU.

<sup>1</sup>We refer to Appendix C of the related paper to execute the program using the functions DeCo\_CPU or DeCo\_GPU and not the GUI.

- The *Panel* "Settings" contains the selection of the "Number of Particles", the "Block of Draws" (see the paper) and the resampling threshold  $\kappa$ . All three options have default values that are also reported. The "Block of Draws" is only relevant for the GPU version and is visible only if a compatible GPU card is present, see Figure 2.
- If the CPU version is run in parallel, the user has to start a parallel session (see also the MATLAB Parallel toolbox users guide) this can be done in two ways:
  - Before using the program (on CPU) open the parallel computing toolbox by adding in the MainPFiterCPU.m the following command: *matlabpool open*;
  - Otherwise type *matlabpool open* directly in the MATLAB command windows before run the DeCo\_GUI.
- The *Panel* "Setting Learning Parameter" allows the user to perform the calculation with or without learning. When the option "Learning" is chosen, the edit box allows to set the learning parameters, the default values are  $\lambda = 0.95$  and  $\tau = 9$ .<sup>2</sup>
- The *Panel* "Set the main diagonal of Lambda and Sigma" reports an editable table that allows the user to set the values for diagonal elements of  $\Sigma$  and  $\Lambda$  (default values are loaded). The check box "Estimate Sigma and Lambda" allows the estimation of the  $\Sigma$  and  $\Lambda$  matrix, once the option is selected the user has to select the "Smoothing factor" ( $h$ , in the paper), a default value of  $h = 0.01$  is used.
- The button "CPU" starts the corresponding CPU program and the button "GPU" executes the program on the GPU. If a compatible GPU is not present then the button GPU and the "Block of Draws" are not visible, see Figure 2. If a compatible GPU is present then the button GPU and the "Block of Draws" are visible, moreover is present a text that gives some hints about the GPU memory available and a suggestion about the range of "Number of Particle" and "Block of Draws" possible.

The program, moreover, gives informations about the GPU card and set a *Static text box* with suggestions using different colors:

- If it has 1 GB of memory, the color is red.
- If it has 2 GB of memory, the color is yellow.
- More then 2 GB, the color is green.

---

<sup>2</sup>The user interested in different scoring rules should change the command lines with the learning in the file PFCoreGPU.m for the GPU version and PFCore.m for the CPU version.

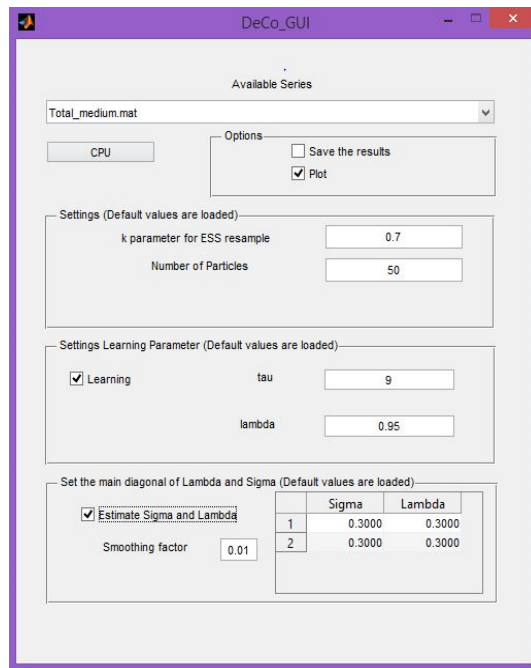


Figure 2: The graphical user interface of the **DeCo** package.

### 3 Dataset format required by the DeCo Toolbox

The user that want to create and run his/her own dataset has to follow the following steps:

- Create a .mat file that has two variables that are defined as follows:
  - The first is defined as  $vY$  and it contains a  $(T \times L)$  matrix of the variables  $y_{1:T}$  to be predicted, where  $T$  is the number of 1-step-ahead forecasts and  $L$  the size of observable variables to forecast;
  - The second is a  $4 - D$  matrix defined  $mX$  with the following dimensions  $(T, M, L, KL)$ , where  $M$  is the size of i.i.d. samples from the predictive densities, and  $KL$  the number of 1-step-ahead predictive densities.
- Copy the file .mat in the folder Dataset.
- Run the program.