

Beyond HMC

A C++ library to estimate HMC and recent models

Stéphane Derrode and Wojciech Pieczynski

January 30, 2010

TABLE OF CONTENTS

1	Hidden Markov Chain Models and recent Extensions	1
1.1	Introduction	1
1.2	Installation	2
1.2.1	Download and install	2
1.2.2	Test installation	3
1.3	Help	5
1.3.1	Simulation programs	5
1.3.2	Restoration programs	6
1.3.3	Estimation and Restoration programs	7
1.3.4	Tool programs	7
1.3.5	Table of program arguments	11
1.4	Tutorial	14
1.4.1	Tutorial #1: Classical HMC model with independent noise and non-gaussian laws	14
1.4.2	Tutorial #2: "PMC model is strictly more general than HMC-CN, which is itself strictly more general than HMC-IN"	17
1.5	Download	21
1.5.1	What's new?	21
1.6	To come in a short (and not so) future!	22
1.6.1	To do list (short term) - before publishing next release (version: 0.3)	22
1.6.2	To do list (middle term) - after the next release (for version: 0.4)	22
1.6.3	To do list (long term)	22
1.7	Bibliography	22
1.7.1	HMC and variants, Pearson, copulas & generalized mixtures, ICE principle...	22
1.7.2	Pairwise Markov chain and HMC with correlated noise	23

1 Hidden Markov Chain Models and recent Extensions

Sources of C++ programs to simulate, restore and estimate hidden Markov chain models including recent extensions.

Site maintained by Stéphane Derrode. Last modified: January 30, 2010, Last program update: January 30, 2010 (version: 0.2).

These pages can also be consulted in pdf format ([html2ps](#) [index.html](#) > [index_v02_20100130.ps](#)).

[Introduction](#) [Installation](#) [Help](#) [Tutorial](#) [Download](#) [To come!](#) [Bibliography](#)

1.1 Introduction

This web site gives access to a collection of programs, written in C++, I have implemented during the last years regarding the so-called Hidden Markov Chain (HMC) model for time-series analysis. Proposed programs are concerned with the simulation, restoration and estimation of recent extensions of HMCs, e.g. Noise-independent HMC (the classical model as described by L. R. Rabiner), Pairwise Markov Chain (including HMC with correlated noise)... See section Bibliography for a set of references about these models.

Most of programs have been developed from the work and with the collaboration of Prof. Wojciech Pieczynki. Bibliographical references regarding the underlying models (and many more!) can be found in its web pages. You can also check for draft papers on my own pages.

These pages are intended only to make the programs available to everyone, in order to ensure maximum dissemination of research results in the scientific community. The code is made freely available for non-commercial use only, especially for research and teaching purposes, provided that suitable citation is made in published work using the functions and programs (site these pages). As a consequence, no responsibility is taken for any errors in the code (but bug report and request is always welcome).

The menu offers several options

Installation	Provide some basic installation instructions and testing commands to check if installation is correct.
Help	Provide a detailed description of available parameters for each supplied programs in order to suit your need for a given application.
Tutorial	Provide examples of experiments you can redo step by step, by following instructions. Most of examples can be used to reproduce figures published in research papers.
Download	Download the latest version of programs. Provide also a detailed description of what has been added from one version of programs with respect to the previous one (backward compatibility with previous versions is not ensure!).
To come!	Provide a to-do list for next versions. Mainly used by me!
Bibliography	Provide detailed bibliographical references for papers cited within these pages, and interesting papers, with link when freely accessible.

Wojciech Pieczynski and Stéphane Derrode



1.2 Installation

The programs are almost self-contained and only require the installation of

GSL library	<i>GNU Scientific Library</i> : furnishes a quite complete collection of numerical algorithms, from random number generation to special function evaluation (and many more). Install packages <i>libgsl0-dev</i> and <i>libgsl0ldbl</i> (using synaptic or <i>apt-get install ...</i> with root privileges).
GnuPlot (optional)	Portable command-line driven graphing utility. This program is used to generate <i>.eps</i> files from results of simulation/restoration. If you do not want to use GnuPlot, comment section regarding Gnuplot in the Makefile's preamble before compiling. Disabling Gnuplot will produce a warning message when using option <i>-x</i> in command line arguments, see Help section.

Programs have been developed in C++, and compile successfully with *gcc version 4.3.2* (under Linux / Debian 4.3.2-1.1). I cannot (and I do not want to!) check for correct installation in other distributions or systems. But programs are expected to work under any environment providing a recent C++ compiler (such as *g++*). If you successfully install programs on specific platforms, please tell me.

1.2.1 Download and install

The last tarred and zipped source code can be found in Download section. For users who have downloaded a previous release and wish to obtain the new functions, it is strongly recommended that the old version be deleted and the entire set downloaded again, to ensure compatibility. After downloading, decompress and unwrap the functions:

```
>> tar -xvzf HMC_Ext_??????.tgz
```

A directory called *HMC_Ext* will be created, in which you can find a *makefile* and 5 directories:

<i>Sources</i>	Source files (.c and .cc files) for classes. They start with an uppercase. Compiling these classes using command <i>make lib</i> generates a library called <i>libHMC_Ext.a</i> in <i>Bin</i> directory. Command <i>make forcelib</i> regenerates entirely the library.
<i>Include</i>	Header files (.h files) for classes. Each header file corresponds to only one class with the same name, and one .cc also with the same name.
<i>Programs</i>	Source files (.cc files) for programs. All programs supplied in the library start with a lower-case: that is to say that programs that start with an upper-case are only for testing and, for that reason, are not documented here). Compiling these programs using command <i>make programs</i> (for all programs) or <i>make programs TARGET=simulHMC-IN</i> (for compiling program <i>simulHMC-IN</i> only) generate binaries in <i>Bin</i> directory.
<i>Bin</i>	Contains the library <i>libHMC_Ext.a</i> and all binary programs generated by compilation. Command <i>make distclean</i> erase all binary files in this directory.
<i>Octave</i>	Contains various scripts to (re)produce figures in Tutorial section.

For short, programs installation gives, first

```
>> make forcelib
      Library ./Bin/libHMC_Ext.a done.
```

for generating the library (it can take a minute), and second

```
>> make programs
      restoreHMC-IN done.
      simulHMC-IN done.
      ...
```

for generating programs.

1.2.2 Test installation

To test correctness of your installation, try to execute command

```
>> ./simulHMC-IN
```

in the *Bin* directory (no text output). This program generates a 2-classes Markov chain file named *X_HMC-IN.txt* and its noisy version named *Y_HMC-IN.txt*. To restore data using both MPM and MAP Bayesian criterion, type command

```
>> ./restoreHMC-IN -f Y_HMC-IN.txt -F X_HMC-IN.txt
      10.6
      10.2
```

Argument for option *-f* provides the file to be restored, whereas argument for option *-F* provides the original Markov chain (generated by the simulation program) for error rate computing after restoration. The two values are the error rates obtained respectively for MPM and MAP criterion (you can expect to get different values due to simulation). Default parameters for restoration are the same that the ones used for simulation, which justify these low error rates. This value can be change using option parameters, see Help section. The restored files are respectively saved in the current directory under names

Y_HMC-IN_ICE_MPM_HMC-IN.txt and *Y_HMC-IN_ICE_MAP_HMC-IN.txt*.

For more versatile programs, use the `-v 1` option

```
>> ./simulHMC-IN -v 1
Simulation of a noisy Markov chain - Independent Noise

*****
./simulHMC-IN [arguments] :
  -Chain length      : 1000
  -Markov matrix    :
0.4  0.1
0.1  0.4

  -X file name      : X_HMC-IN.txt
  -Y file name      : Y_HMC-IN.txt
  - Law:Gaussian    (8); mu1,mu2,mu3,mu4 : +0.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Law:Gaussian    (8); mu1,mu2,mu3,mu4 : +2.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  -Verbose level    : 1
*****

Number of data with class 0 in the simulated chain : 533, i.e. 53.3%

Time spend during following functions (in sec.):
Simulation time.....0

End of - Simulation of a noisy Markov chain - Independent noise
```

and

```
>> ./restoreHMC-IN -f Y_HMC-IN.txt -F X_HMC-IN.txt -v 1
Estimation+restoration of a noisy Markov chain - Independent noise

*****
./restoreHMC-IN [arguments] :
  -Markov matrix    :
0.4  0.1
0.1  0.4

  -Y file name      : Y_HMC-IN.txt
  -Orig. Markov file: X_HMC-IN.txt
  - Law:Gaussian    (8); mu1,mu2,mu3,mu4 : +0.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Law:Gaussian    (8); mu1,mu2,mu3,mu4 : +2.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  -Verbose level    : 1
*****

The error rate for MPM is 12.2
The error rate for MAP is 12.9

Time spend during following functions (in sec.):
Restoration time for MPM.....0
Restoration time for MAP.....0

End of - Restoration of a noisy Markov chain - Independent noise
```

Spent time only counts for scientific computation, not for file reading and/or saving.

Also, you can use `-h` options, to get a list of available parameters for a given program. For example:

```
>> ./simulHMC-IN -h
./simulHMC-IN [options]
-n Chain lenght [>1]; default=1000
-m Joint Markov proba; default=0.4,0.1,0.1,0.4
-T Parameters for law #1:
  law in [0=Gaussian; 1=Gamma; 2=8_2_7 in Pearson system; 3= case 2 + case 1; 4=all Pearson],
  sign(+1,-1), mean, var>0, betal>=0, beta2>=1; default=0,1,0,1,0,3
-U Parameters for law #2:
  law in [0=Gaussian; 1=Gamma; 2=8_2_7 in Pearson system; 3= case 2 + case 1; 4=all Pearson],
  sign(+1,-1), mean, var>0, betal>=0, beta2>=1; default=0,1,2,1,0,3
-f Filename X,Y; default=X_HMC-IN.txt,Y_HMC-IN.txt
-v Verbose [0/1/2]; default = 0
-h This help!
-F Filename of a binary Markov chain => -m not required
-x Save graphic files to a specified directory
```

To get explanation on argument values for each program, see the lists reported in Help menu.

1.3 Help

Remark: I only give here some basic instructions regarding use of programs, not provide detailed documentation or a comprehensive software support service! News of any errors found however will be gratefully received.

This page provides explanations on all arguments you can provide to programs, e.g. `-e` or `-f`. Arguments keep their meaning in all programs, so you should be able to quickly learn common argument options. It is always possible to pass argument `-h` to a program to get a comprehensive text list of possible arguments for that program. Each argument has a default value, which should be enough to learn and make basic simulations.

There is 4 kinds of programs:

- Simulation programs, to generate a noise-free and a noisy sample from a Markov model,
- Restoration programs, to restore a noisy signal from model parameter values provided by user,
- Estimation+restoration programs, to restore a noisy signal from model parameter values estimated by using iterative routine (most of the time I implemented ICE *Iterated Conditional Estimation*, but sometimes EM and SEM are available), and
- Tool programs, to compute an error rate between two binary-valued files, or to get a figure of a density from the Pearson' system of distributions...

Remark 1: Just click on an argument to get information on it.

Remark 2: All programs are limited to 2 classes (binary Markov chain), for ease of use. This is not a theoretical limitation, since all models are valid for a finite number of classes.

Remark 3: Program options are presented with their default value, which has been choose to be the simplest (e.g. always product copulas are chosen by default, whereas more interesting copulas, such as Gaussian and Student' copulas are available). Do not hesitate to modify the default choices!

1.3.1 Simulation programs

Simulations programs refer to routines that simulate data according to a given Markov model. Generated data can be use with restoration and estimation programs to evaluate their capacity to deal with data following exactly the underlying model.

Simulation of data following an HMC-IN model

```
>> ./simulHMC-IN -n 1000 -m 0.4,0.1,0.1,0.4 -T 0,1,0,1,0,3 -U 0,1,2,1,0,3 -f X_HMC-IN.txt,Y_HMC-IN.txt
-v 0
```

Three other options are available: -F, -x and -h.

Simulation of data following a PMC model

```
>> ./simulPMC -n 1000 -m 0.4,0.1,0.1,0.4 -c 1,1,1,1 -T 0,1,0,1,0,3 -t 0,1,0.3,1,0,3 -u 0,1,1.7,1,0,3 -U
0,1,2,1,0,3 -v 0
>> ./simulPMC -n 1000 -m 0.4,0.1,0.1,0.4 -c 1,1,1,1 -T 0,1,0,1,0,3 -U 0,1,2,1,0,3 -p -v 0
```

Simulation of PMC model can be degenerated to the simulation of HMC-CN (Correlated Noise) model using option -p (second example). In this case, only options -T and -U are taken into account (i.e. not options -t and -u).

A list of available copulas can be printed using option -z. If selected copulas require parameters (e.g. Gaussian copula require correlations coefficients), then values are asked to the user during program execution. Please note that copula and parameters for $(x_n=0, x_{n+1}=1)$ and $(x_n=1, x_{n+1}=0)$ should be the same, due to some symmetry condition required by the model.

Three other options are available: -F, -x and -h.

1.3.2 Restoration programs

The restoration programs deal with the restoration of a noisy signal according to a Markov model for which user provides parameter values. The programs apply, in most case, both MPM and MAP criteria, and save the restored chain on the disk. Error rate is computed only if the original data to be compared with the restored signal is provided using -F argument (you can always compute the error rate between text files using the *erroRate* tool program, see below).

Restoration using an HMC-IN model from provided parameters

```
>> ./restoreHMC-IN -m 0.4,0.1,0.1,0.4 -T 0,1,0,1,0,3 -U 0,1,2,1,0,3 -f Y_HMC-IN.txt -v 0
```

Three other options are available: -F, -x and -h.

Restoration using a PMC model from provided parameters

```
>> ./restorePMC -m 0.4,0.1,0.1,0.4 -c 1,1,1,1 -T 0,1,0,1,0,3 -t 0,1,0.3,1,0,3 -u 0,1,1.7,1,0,3 -U 0,1,2,1,0,3 -f
Y_PMC.txt -v 0
>> ./restorePMC -m 0.4,0.1,0.1,0.4 -c 1,1,1,1 -T 0,1,0,1,0,3 -U 0,1,2,1,0,3 -f Y_PMC.txt -p -v 0
```

Restoration of PMC model can be degenerated to the restoration of HMC-CN (Correlated Noise) model using option -p (second example). In this case, only options -T and -U are taken into account (i.e. not options -t and -u).

A list of available copulas can be printed using option -z. If selected copulas require parameters (e.g. Gaussian copula require correlations coefficients), then values are asked to the user during program execution. Please note that copula and parameters for $(x_n=0, x_{n+1}=1)$ and $(x_n=1, x_{n+1}=0)$ should be the same, due to some symmetry condition required by the model.

Three other options are available: -F, -x and -h.

1.3.3 Estimation and Restoration programs

The estimation and restoration programs deal with the restoration of a noisy signal according to a Markov model for which parameters are estimated using an iterative procedure (most of the time, the ICE procedure is used, but sometimes EM and SEM are also proposed as alternatives or for comparison purposes). The program applied, in most case, bot MPM and MAP criteria, and save the restored chain on the disk. Error rate is computed only if the original data to be compared with the restored signal is provided using -F argument (you can always compute the error rate between text files using the *erroRate* tool program, see below).

Estimation and restoration using an HMC-IN model

```
>> ./estimHMC-IN -E ICE -e 50 -l 0,0 -f Y_HMC-IN.txt -v 0
```

Three other options are available: -F, -x and -h.

Estimation and restoration using a PMC model

```
>> ./estimPMC -E ICE -e 50 -c 1,1,1,1 -l 0,0,0,0 -f Y_PMC.txt -v 0
>> ./estimPMC -E ICE -e 50 -c 1,1,1,1 -l 0,0,0,0 -f Y_PMC.txt -p -v 0
```

Estimation of PMC model can be degenerated to the estimation of HMC-CN (Correlated Noise) model using option -p (second example). In this case, option -l always takes a list of 2 integer, but only the first and fourth integer are used to specify the law type for the 2 margins.

A list of available copulas can be printed using option -z

Three other options are available: -F, -x and -h.

1.3.4 Tool programs

Here is a collection of programs not directly related to Markov modelling but of some interest for display, error rate computation...

Error rate between two binary files

```
>> ./errorRate -i X_HMC-IN.txt,Y_HMC-IN_ICE_MAP_HMC-IN.txt -v 0
```

One other option is available: -h.

Draw one or two densities from the Pearson system of distributions

```
>> ./drawPearsonPdf -T 4,1,-10,10,0,4 -U 1,-1,5,20,1.5,4 -v 0 -x ./
```

One other option is available: -h. By default, option -x is set to ./.

Draw a 2D pdf from the specification of its 2 margins (within Pearson' system) and its copula

```
>> ./drawCopula2D -c 3 -T 4,1,11,6,0.7,4.3 -U 4,1,10,5,3,9.5 -v 1 -x ./
```

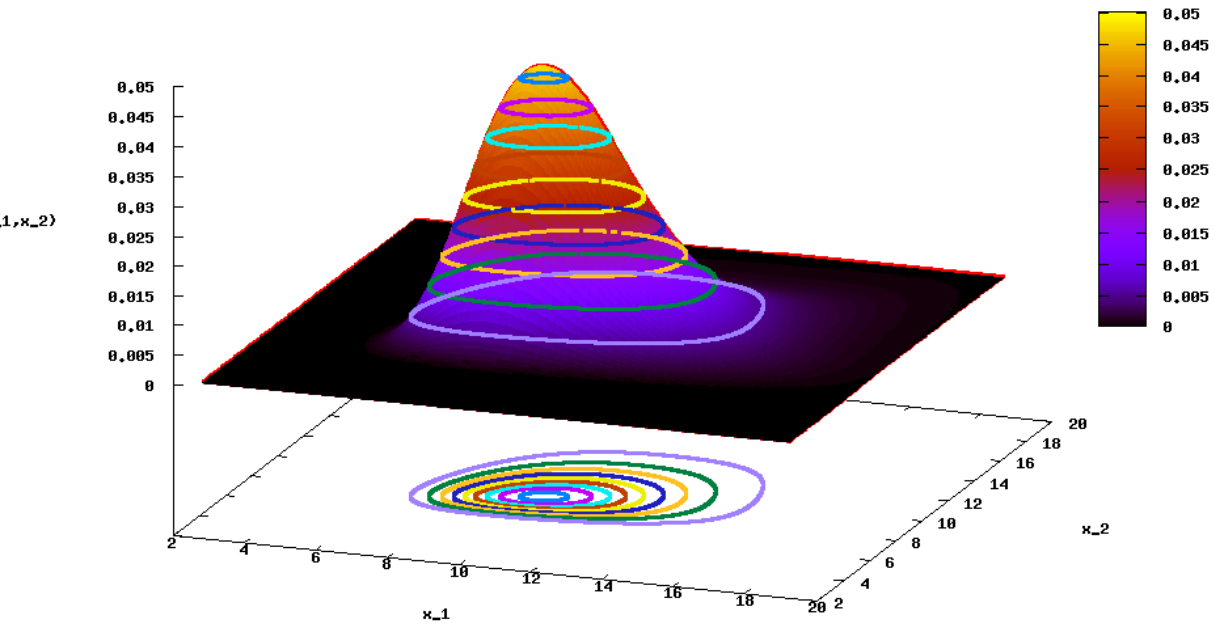
This program draws a 2D pdf built with a Student copula (correlation coeff is specified during program execution) with 2 degrees of freedom (fixed in the program), with a Gaussian margin (as specified by -T option) and a second kind Beta margin (as specified by -U option). It takes a few seconds to generate the figures. By default, option -x is set to ./.

The example above gives the following display:

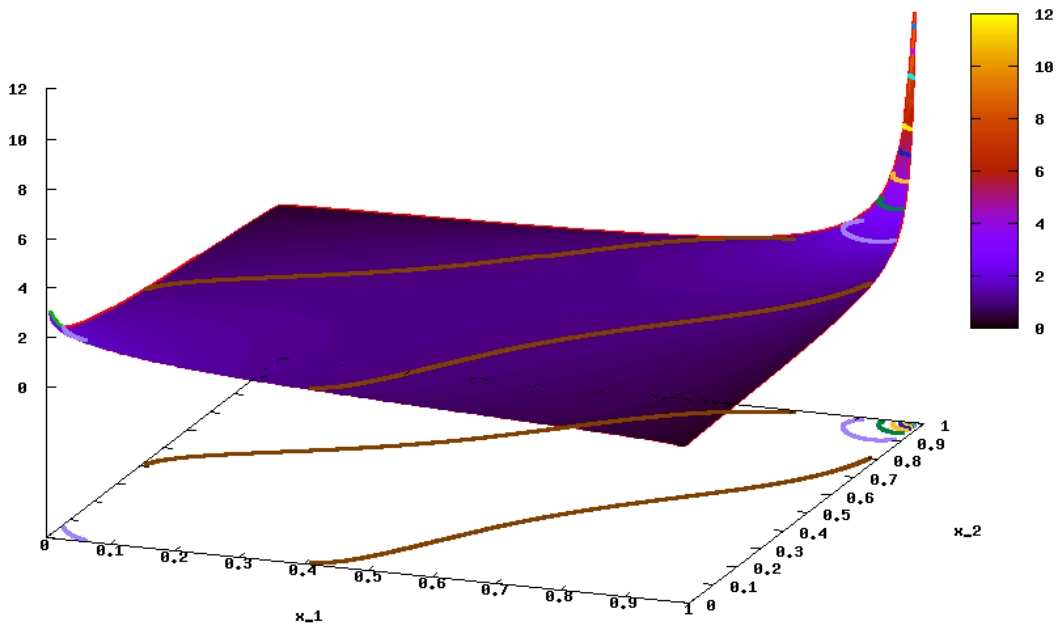
```
*****
./drawCopula2D [arguments] :
-Copula type and margins :
  ->Copula : Gumbel-Hougaard - Theta=0.8
  ->Margins
    - Beta 2; mu1,mu2,mu3,mu4 : +11.00,+6.00,+12.30,+154.80, beta1,2 = +0.70 & +4.30
    - Type V; mu1,mu2,mu3,mu4 : +10.00,+5.00,+19.36,+237.50, beta1,2 = +3.00 & +9.50
-Verbose level      : 2
-Graphic files to  : .
*****
```

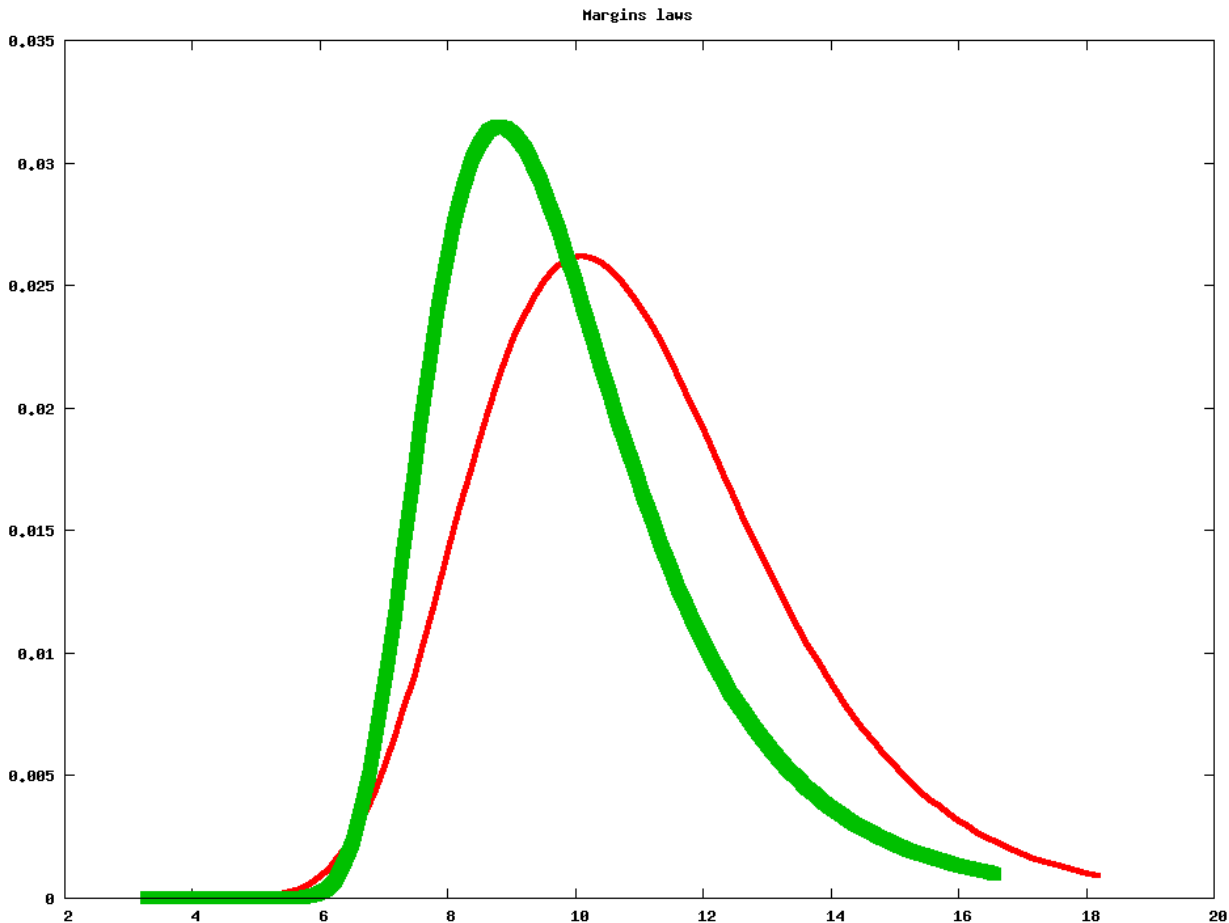
and generate the following figures:

Pdf with specified copula and margins



Copula shape





Two other options are available: -h and -z.

1.3.5 Table of program arguments

-n #integer value>1 default: 1000	Simulated chain length. Should be not too small to allow statistical estimation and not too large for a few programs (computing load).
-e #integer value>=0 default: 50	Number of iterations used by iterative routines for parameters estimation. There is no stop rule based on parameters evolution during iterating process.
-E string ICE, SEM or EM default:ICE	Allow to select the method used to estimate parameters. For PMC, EM is nos implemented so that the choice is reduced to ICE and SEM.
-m list of reals>0 default: 0.4,0.1,0.1,0.4	Stationary Markov matrix. The list of values gives M(0,0), M(0,1), M(1,0), M(1,1). If M is not sum-to-one, a normalization is performed.

<p>-l list of integer each value in [0,4] default: 0,0,...</p>	<p>Allow to specify the sub-set of Pearson' system of distributions for each data-driven density:</p> <ul style="list-style-type: none"> ● #0: Gaussian law ● #1: Gamma law ● #2: Law 8, 2 and 7 ● #3: Options 2 and 1 ● #4: All distributions in Pearson' system <p>The number of laws depends on the program (see option -h of a program to know the desired number of laws). When used in estimation programs, this option restrict the search in the Pearson' system of distributions to the specified sub-family. Explanations on the Pearson' system of distributions is provided here (pdf, 20 pages, in French, sorry!)</p>
<p>-T list of reals -U list of reals -t list of reals -u list of reals default: -T 0,1,0,1,0,3 -U 0,1,2,1,0,3 -t 0,1,0.3,1,0,3 -u 0,1,1.7,1,0,3</p>	<p>These arguments define the parameters needed to specify laws in the Pearson' system of distributions. Parameters are:</p> <ul style="list-style-type: none"> ● law: same argument as option -l ● sign: sign of parameter $\gamma_1 = \sqrt{\beta_1}$ in [+1, -1] ● mean: mean value ● var: variance value >0 ● beta1: skewness value >0 ● beta2: kurtosis value >1 <p>For law in [0,3], a correction is performed on subsequent parameters in order to fit the desired family of laws. Definition of coefficients are provided here.</p>
<p>-f string default: X_HMC-IN.txt, Y_HMC-IN.txt</p>	<p>Filenames for the Markov chain X and its noisy version Y. This option is used for writing as well as for reading files.</p>
<p>-i string no default value</p>	<p>Depending on the context, this argument is used to provide some files (separated by commas) to the program. This option is only used for tool programs.</p>

<p>-F string no default value</p>	<p>This option is used to provide a filename to a program. The expected file has a different meaning depending on context. For examples:</p> <ul style="list-style-type: none"> • To provide the original binary chain for computing error rates when using restoration and estimation programs. • To inform simulation programs to only simulate noisy observations, and considering the provided file as a realization of the expected Markov chain.
<p>-v #integer value in [0,2] default: 0</p>	<p>Verbose level during program execution:</p> <ul style="list-style-type: none"> • Value 0 means that very few information are given (in general, only error rates). • Value 1 means that only important information are reported. • Value 2 is for debugging.
<p>-h no value</p>	<p>Alone, to get a complete list of possible arguments for the program.</p>
<p>-z no value</p>	<p>Alone, to get a complete list of possible copulas for the program. Note that the list can be different from one program to the other.</p>
<p>-p no value</p>	<p>To use an HMC-CN (Correlated Noise) model in place of a PMC. In that case, only margins characterize with options -T and -U are used.</p>
<p>-c list of integers default: 1,...</p>	<p>To set the list of copulas needed for the program. The number of copulas required can be different from one program to the other (e.g. 4 copulas for a binary PMC model, and only one copula for the program called <i>drawCopulas2D</i>). Integer value should be chosen between 0 and 5:</p> <ul style="list-style-type: none"> • #0 -> Product Copula • #1 -> Gaussian Copula • #2 -> Student' Copula • #3 -> Logistic Gumbel Copula • #4 -> Farlie-Gumbel-Morgenstern Copula • #5 -> Cubic section Copula <p>Be aware that available copulas can be different from one program to the other. Use option -z to get a list of available copulas for the program of interest. Of course, you can mix copulas, e.g -c 0,1,2,3, but you can expect very strange results!.</p>

<pre>-x string no default value</pre>	Set the repositories where <i>.eps</i> files will be saved. it concerns drawings about signal, histograms, pdf, copulas... In some programs, figures generation is quite time consuming. Using this option whereas Gnuplot section has been commented in Makefile generates a warning message.
---------------------------------------	--

1.4 Tutorial

Here is a collection of, simple or more sophisticated, examples to illustrate possible uses of programs. This list will growth each time a new model (or variant) is integrated.

Some reported plots are generated using *Octave* functionalities, from the saved text files. Octave scripts are provided in the *Octave* directory. Calls to scripts are also reported for ease of reproduction.

Other plots are directly generated by Gnuplot during execution using *-x* option (assuming that programs have been compiled with Gnuplot enable, see Installation section).

List of tutorials:

- Tutorial #1: Classical HMC model with independent noise but non-gaussian laws
- Tutorial #2: Illustration of "PMC model is strictly more general than HMC-CN, which is itself strictly more general than HMC-IN"

1.4.1 Tutorial #1: Classical HMC model with independent noise and non-gaussian laws

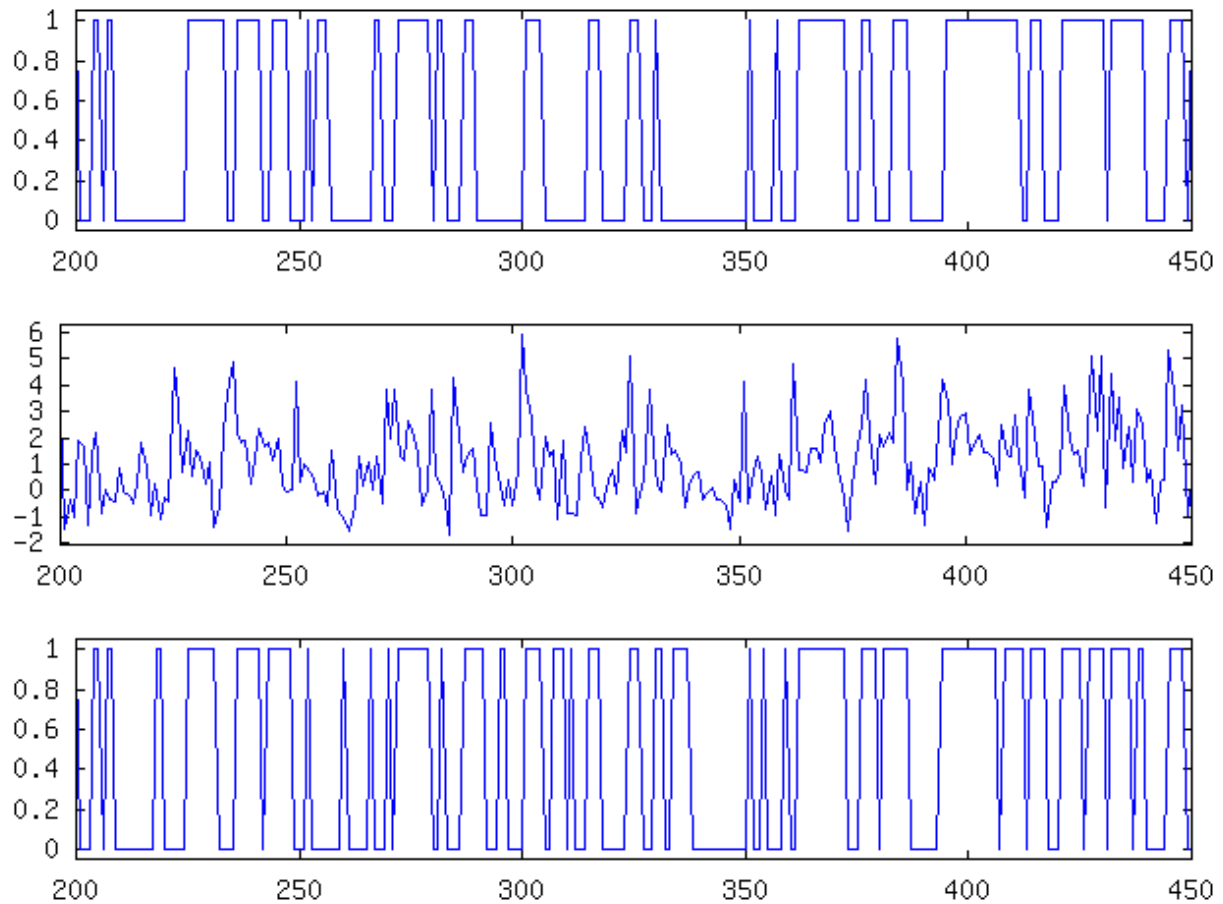
Simulation of an HMC-IN mixture of two gamma laws

```
>> ./simulHMC-IN -n 10000 -m 0.4,0.1,0.1,0.4 -T 1,1,0,1,1,4.5 -U 1,1,2,2,2,6 -f
X_HMC-IN.txt,Y_HMC-IN.txt -v 1
```

and estimation using 100 ICE iterations

```
>> ./estimHMC-IN -E ICE -e 100 -l 1,1 -f Y_HMC-IN.txt -v 1 -F X_HMC-IN.txt -x ./
```

This gives MPM and MAP error rates of 11.25% and 11.89% respectively. Next figure shows the simulated Markov chain (up), the simulated observed data (middle) and the ICE estimation and MPM restoration (down).



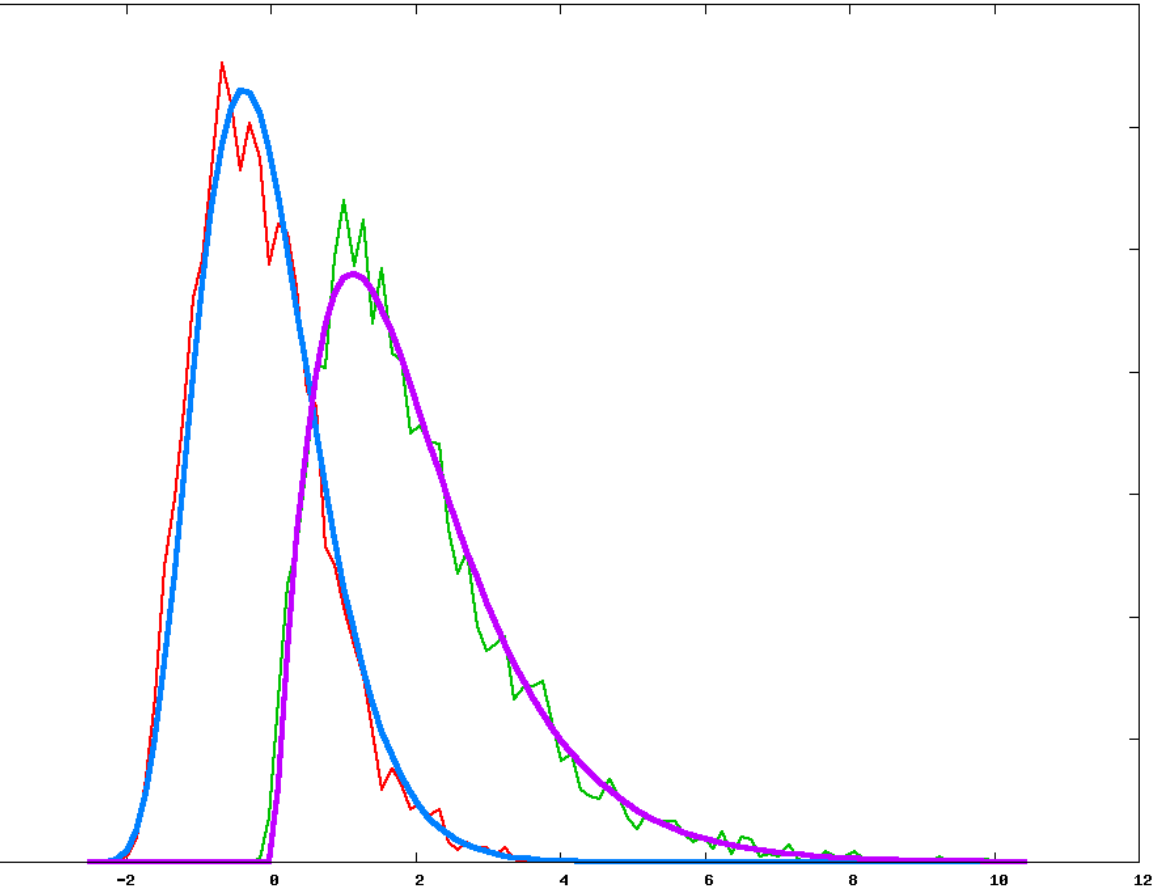
Here is the call to script *Example1.m* used to generate this figure (from Octave directory):

```
>> Example1(200, 450, './Bin/X_HMC-IN.txt', './Bin/Y_HMC-IN.txt',
 './Bin/Y_HMC-IN_ICE_MPM_HMC-IN.txt')
```

The first two arguments define the extent of the window to be plotted (samples index between 200 and 450).

The `-x` option can be used to set the repository where graphic files (with `.eps` extension) will be saved. For program `./estimHMC-IN` we get the one figure which superposes the histogram of classes and estimated densities after classification.

Normalized Histograms and Estimated Pdf - All classes



After 100 ICE iterations, estimated parameters for the two laws are

- Gamma (3); $\mu_1, \mu_2, \mu_3, \mu_4$: +0.02, +1.05, +1.05, +4.88, $\beta_{1,2} = +0.96$ & +4.43 (Sample size: 4958)
- Gamma (3); $\mu_1, \mu_2, \mu_3, \mu_4$: +2.01, +2.06, +4.00, +24.41, $\beta_{1,2} = +1.82$ & +5.74 (Sample size: 5042)

whereas Markov joint a priori matrix is given by

$$\begin{pmatrix} 0.401937 & 0.0938484 \\ 0.09378 & 0.410434 \end{pmatrix}$$

with a priori probabilities:

$$\begin{pmatrix} 0.495786 \\ 0.504214 \end{pmatrix}$$

and transition matrix:

$$\begin{pmatrix} 0.810708 & 0.189292 \\ 0.185992 & 0.814008 \end{pmatrix}$$

Estimated parameters are close to the parameters used for simulation.

1.4.2 Tutorial #2: "PMC model is strictly more general than HMC-CN, which is itself strictly more general than HMC-IN"

Here is a set of experiments that illustrates the sentence, by first generating a PMC signal and then restoring it with PMC, HMC-CN and HMC-IN models.

We first generate a signal of 100000 samples following a PMC model with default parameters for margins (i.e. Gaussians) and copula (i.e. Gaussians):

```
>> ./simulPMC -n 100000
```

Correlation parameters for the 4 copulas are respectively set to 0.2 for $(x_n=0, x_{n+1}=0)$, 0.5 for $(x_n=0, x_{n+1}=1)$, 0.5 for $(x_n=1, x_{n+1}=0)$ and 0.7 for $(x_n=1, x_{n+1}=1)$. Supplementary option -v 2 gives the following display:

```
- Markov parameters :
  0.4 0.1
  0.1 0.4
- Laws
  (0,0)
  ->Copula : Gaussian -
    1    0.2
    0.2  1
  ->Margins
  - Gaussian; mu1,mu2,mu3,mu4 : +0.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Gaussian; mu1,mu2,mu3,mu4 : +0.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  (0,1)
  ->Copula : Gaussian -
    1    0.5
    0.5  1
  ->Margins
  - Gaussian; mu1,mu2,mu3,mu4 : +0.30,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Gaussian; mu1,mu2,mu3,mu4 : +1.70,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  (1,0)
  ->Copula : Gaussian -
    1    0.5
    0.5  1
  ->Margins
  - Gaussian; mu1,mu2,mu3,mu4 : +1.70,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Gaussian; mu1,mu2,mu3,mu4 : +0.30,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  (1,1)
  ->Copula : Gaussian -
    1    0.7
    0.7  1
  ->Margins
  - Gaussian; mu1,mu2,mu3,mu4 : +2.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
  - Gaussian; mu1,mu2,mu3,mu4 : +2.00,+1.00,+0.00,+3.00, beta1,2 = +0.00 & +3.00
```

The two generated files *X_PMC.txt* and *Y_PMC.txt* contains the simulated PMC.

We next restore the signal with the PMC, the HMC-CN and the HMC-IN models (when required we set the correlation parameters with the values above):

```
>> ./restorePMC -F X_PMC.txt
```

which gives a MPM error rate of 14.803% and a MAP-one of 15.449%.

```
>> ./restorePMC -F X_PMC.txt -p
```

which gives a MPM error rate of 15.342% and a MAP-one of 16.424%.

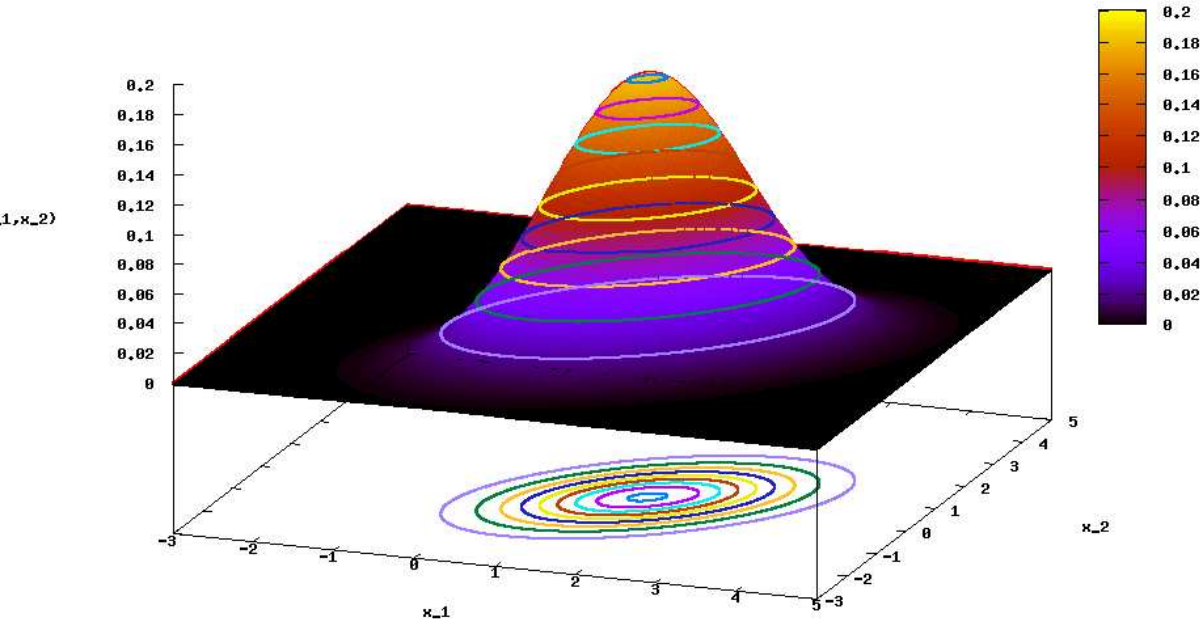
```
>> ./restoreHMC-IN -f Y_PMC.txt -F X_PMC.txt
```

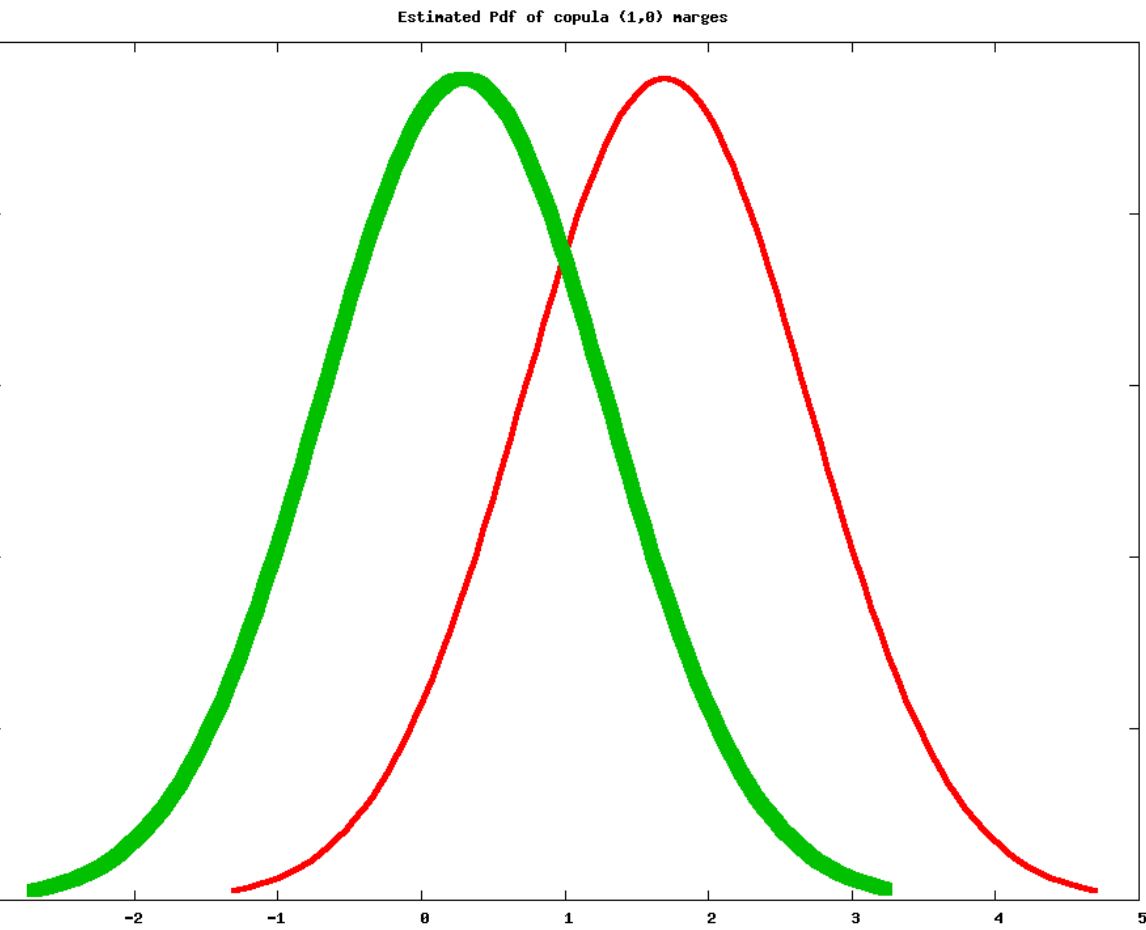
which gives a MPM error rate of 17.341% and a MAP-one of 17.811% (we must specify option -f since, by default, the program looks for a file named *Y_HMC-IN.txt*).

Difference between restoration rates can be accentuated by playing with parameters (e.g. Markov matrix, correlation value for copulas and non Gaussian laws). To confirm the higher generality of the PMC model vs HMC-IN ones, we can do the inverse experiment: simulating an HMC-IN signal and restoring it with the three models (setting all correlation parameters to 0): error rates are very similar.

Using option -x ./ in *restorePMC* program generates postscript figures of marge densities and copula densities (of course, if gnuplot is installed). For $(x_n=1, x_{n+1}=1)$, we get the following draws:

Estimated Pdf of copula (1,0)

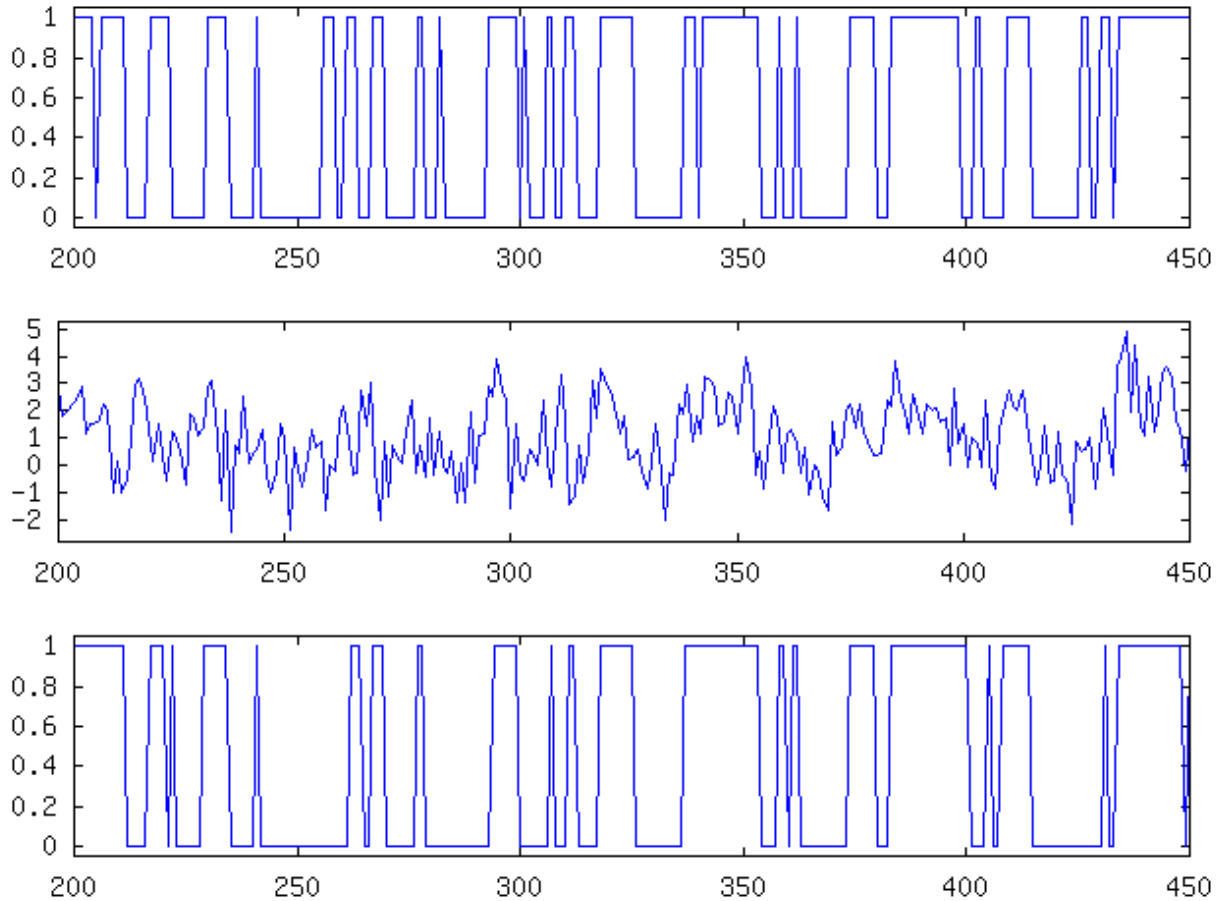




Signals can be observed with a call to octave script *Example1.m* :

```
>> Example1(200, 450, '../Bin/X_PMC.txt', '../Bin/Y_PMC.txt', '../Bin/Y_PMC_ICE_MPM_PMC.txt')
```

we get



1.5 Download

1.5.1 What's new?

For users who have downloaded a previous release and wish to obtain the new functions, it is strongly recommended that the old version be deleted and the entire set downloaded again, to ensure compatibility.

All versions of programs can be accessed [here](#).

HMC_Ext_20100130.tgz (version 0.2 - 100ko)

Add three programs for the simulation, restoration and estimation of pairwise Markov chains [B1, B2]. By default, programs work with Gaussian densities for margins and Gaussian copulas. Using provided options, you can select different copulas (Gaussian, Student, Logistic...) and margins inside the Pearson' system of distributions [A10, A11]. Available copulas are (till now!): Product, Gaussian, Student, cubic section, Gumbel logistic and Farlie-Gumbel-Morgenstern (FGM) [A12, A13, A14].

Option to deal with the HMC-CN (Correlated Noise) model --a particular case of PMC which is also strictly more general than the HMC-IN model-- is available.

By default, model parameters estimation is performed using the ICE principle, but SEM is also available for comparison purpose. We also provided the HMC-IN model with EM and SEM parameters estimation

procedures for comparison purposes.

Three tool programs have also been added. The first one for the computation of an error rate between two binary files. The second can be used to draw one or two densities from the Pearson' system of distributions, providing their moments. The third one to draw a 2D density with specified copula (from the available list) and margins (from the Pearson' system).

Be aware that some of programs in version 0.1 (and their arguments) have been modified. Compatibility between old and new programs (as well as their behavior) is not ensure!

HMC_Ext_20091220.tgz (version 0.1 - 63ko)

First deposite. Propose three programs for the simulation, restoration and estimation of the HMC-IN (Independent Noise) model described by L. R. Rabiner [A1]. By default, data-driven densities are Gaussian. But arguments provide a way to deal with more sophisticated densities coming from the Pearson' system of distributions [A10]. Model parameters estimation is performed using the ICE principle [A2, A4, A5, A6].

1.6 To come in a short (and not so) future!

1.6.1 To do list (short term) - before publishing next release (version: 0.3)

- Extend estimation programs (*estimHMC-IN* and *estimPMC*) to manage more than $K=2$ classes.
- Make two small programs to allow image segmentation (using the Hilbert-Peano decomposition). The idea is to "signalize" the image before processing and to "designalize" the segmented chain to recover the segmented images.
- Develop a set of programs (simulation, restoration and estimation-restoration) to deal with independent (i.e. non-markavian) data.

1.6.2 To do list (middle term) - after the next release (for version: 0.4)

- Make small tcl-tk based graphical interfaces which propose options to user and generates automatically the corresponding command line.
- HMC model with local and bootstrap estimation (eventually extend the method to the PMC mdoel)

1.6.3 To do list (long term)

- Fuzzy HMC model
- Higher-order HMC model
- Long Memory HMC model

1.7 Bibliography

Here is a collection of key bibliographical references cited inside these pages. It should not be considered as an exhaustive bibliography! These papers are of particular interest to readers interested in theoretical justifications. When available, a link is provided to download preprint.

1.7.1 HMC and variants, Pearson, copulas & generalized mixtures, ICE principle...

- [A1] L. R. Rabiner, *A tutorial on hidden Markov model and selected applications in speech recognition*, Proc. of the IEEE, 77(2), 257-286, Feb. 1989. Link
- [A2] B. Benmiloud, W. Pieczynski, *Estimation des paramètres dans les chaînes de Markov cachées et segmentation d'images*, Traitement du Signal, 12(5), 433-454, 1995. Link

- [A3] N. Giordana, W. Pieczynski, *Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(5), 465-475, 1997. Link
 - [A4] W. Pieczynski, *Champs de Markov cachés et estimation conditionnelle itérative*, Traitement du Signal, 11(2), 141-153, 1994. Link
 - [A5] W. Pieczynski, *Sur la convergence de l'estimation conditionnelle itérative*, Comptes Rendu de l'académie des Sciences-Mathématique, 346(7-8), 457-460, Avril 2008. Link
 - [A6] J.-P. Delmas, *An equivalence of the EM and ICE algorithm for exponential family*, IEEE Trans. Signal Processing, 45(10), 2613-2615, 1997. Link
 - [A7] W. Pieczynski, J. Bouvrais, C. Michel, *Estimation of generalized mixture in the case of correlated sensors*, IEEE Trans. on Image Processing, 9(2), 308-311, 2000. Link
 - [A8] Y. Delignon, A. Marzouki, W. Pieczynski, *Estimation of generalized mixture and its application in image segmentation*, IEEE Trans. on Image Processing, 6(10), 1364-1375, 1997. Link
 - [A9] S. Derrode, G. Mercier, *Multiscale oil slick segmentation from SAR image using a vector HMC model*, Pattern Recognition, 40(3), 1135-1147, March 2007. Link
 - [A10] N. L. Johnson, S. Kotz, *Distribution in statistics: Continuous univariate distributions*, Vol. 1 and 2. New York. John Wiley and Sons, 1994.
 - [A11] S. Derrode, *Description du système de Pearson pour son implémentation en langage C à l'aide de la librairie GSL*. Rapport interne (in French), Ecole Centrale Marseille & Institut Fresnel, 2009. Link
 - [A12] T. Roncalli, *Gestion des Risques Multiples ou Copules et Aspects Multidimensionnels du Risque*. Cours ENSAI de 3ième année (in French), Groupe de Recherche Operationnelle - Crédit Lyonnais, 233 pages. Link
 - [A13] G. Mercier, *Mesures de dépendance entre images RSO*. Rapport de recherche (in French), RR-2005003-ITI, Octobre 2005. Télécom Bretagne & TAMCIC (CNRS UMR 2872), Equipe TIME, 2009. Link
 - [A14] L. Benyoussef, C. Carincotte, S. Derrode, *Extension of higher-order HMC modeling with application to image segmentation*, Digital Signal Processing, Vol. 18(5), pp. 849-860, 2008, doi: 10.1016/J.dsp.2007.10.010. Link
-

1.7.2 Pairwise Markov chain and HMC with correlated noise

- [B1] N. Brunel, W. Pieczynski, *Unsupervised signal restoration using hidden Markov chains with copulas*, Signal Processing, 85(12), 2304-2315, 2005. Link
- [B2] S. Derrode, W. Pieczynski, *Signal and image segmentation using pairwise Markov chains*, IEEE Trans. on Signal Processing, 52(9), 2477-2489, 2004. Link
- [B3] W. Pieczynski, *Pairwise Markov chains*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(5), 634-639, 2003. Link
- [B4] N. Brunel, *Sur quelques extensions des chaînes de Markov cachées et couples. Application à la segmentation non supervisée des signaux radar.*, thèse de l'Université Paris VI (in French), soutenue le 5 décembre 2005. Encadrants: Frédéric Barbaresco (Thales Air Defense), Paul Deheuvels (Paris VI, LSTA), et Wojciech Pieczynski CITI, INT. Link