

Beyond HMC

Stéphane Derrode and Wojciech Pieczynski

Hidden Markov Chain Models and recent Extensions

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

If you are looking for the on-line demos, go [there](#).

Site maintained by [Stéphane Derrode](#). Last modified: November 06, 2011, Last program update: November 06, 2011 (version: 0.7). These pages can also be consulted in [pdf](#) (generated using `htmldoc`, 1.4Mo).

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

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. Programs are concerned with the simulation, restoration and estimation of recent extensions of HMC models, e.g. Noise-independent HMC (the classical model as described by L. R. Rabiner [A1]), Pairwise Markov Chain (including HMC with correlated noise [B1, B2, B3]), Triplet Markov Chain (the simple and less simple ones...). 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 Pieczynski](#). 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 [personal pages](#).

This site is only intended to make programs available to everyone, in order to ensure a wide 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 (please cite these pages). As a consequence, no responsibility is taken for any errors in the code (but bug report and request are always welcome).

The menu offers several options

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 ensured!).
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.
Bibliography	Provide detailed bibliographical references for papers cited within these pages, and interesting papers, with link when freely accessible.
To come!	Provide a to-do list for next versions. Mainly used by me (not to forget to work again and again;-))!

Stéphane Derrode



Download

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

What's new?

[HMC_Ext_20111106b.tgz](#) (version 0.7 - 225ko)

Nothing really new, but many improvements regarding copulas, robustness... Triplet Markov chains (version 0.6) have been suppressed since TMC programs require more tests and more robustness. Programs *estimBLIND*, *estimHMC-IN* and *kmeans* now support multiple input files (example: `-Y R.txt:G.txt:B.txt`). Data driven densities are modeled using copulas that can be chosen in a set of 8 available copulas (product, Gaussian, Student, Gumbel-Hougaard, Cubic section...) and margins within the Pearson' system of distributions. Only *plplot* is supported for graphics (*gnuplot* deprecated).

[HMC_Ext_20101101.tgz](#) (version 0.6, no more available)

In this new version, we added three programs to simulate, restore and estimate (using ICE or SEM) triplet Markov chain data. For example, you can now write

```
>> ./simulTMC
>> ./restoreTMC -F X_TMC.txt:U_TMC.txt
>> ./estimTMC -F X_TMC.txt:U_TMC.txt
```

to get the error rate against the true parameters (*restoreTMC*), and the error rate obtained with a 50-iterations SEM-based parameters estimation (*estimTMC*). The only new parameter is the number of classes in the latent process U, which set the number of stationarities in the data. A tutorial on TMC model will be published with the next release (in which a more general TMC model should be included).

[HMC_Ext_20101014.tgz](#) (version 0.5b - 191ko)

In this (slight) new version, we only change the way to select a subset of laws from the Pearson system:

```
0: Gaussian
1: Gamma
2: Student
3: Gamma inverse
4: Beta first kind
5: Beta second Kind
6: All Pearson' system
```

[HMC_Ext_20101012.tgz](#) (version 0.5 - 170ko)

In this new version, we added three programs to simulate, restore and estimate (using ICE, EM and SEM) i.i.d. data. For example, you can now write

```
>> ./simulBLIND
>> ./restoreBLIND -F X_BLIND.txt
>> ./estimBLIND -F X_BLIND.txt
```

Beyond HMC

to get the error rate with the true parameters (*restoreBLIND*), and the error rate with 300-iterations EM-based parameters estimation (*estimBLIND*). The main objective of these programs is for comparison with Markovian models.

[HMC Ext 20100722.tgz](#) (version 0.4b - 151ko)

In this (slight) new version, commands parsing (i.e. control of program options) have been made more robust for main programs. For example, you can now write

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

and default option values will be considered. You can also write

```
>> ./simulHMC-IN -i InitHMC-IN.txt
```

and options in file *InitHMC-IN.txt* will replace default values. The better way to write such a file is to modify [this file](#) for all programs regarding the HMC-IN model (i.e. *simulHMC-IN*, *restoreHMC-IN*, *estimHMC-IN* and *estimHMC-IN_BL*) and to modify [this file](#) for all programs regarding the PMC model (i.e. *simulPMC*, *restorePMC* and *estimPMC*). The last way to send arguments to a program is to include options manually like this

```
>> ./simulHMC-IN -i InitHMC-IN.txt -v 2
```

Options given manually will replace all previous values. By adding *-h* in the sentence the displayed help will show all actual values considered in program call.

[HMC Ext 20100714.tgz](#) (version 0.4 - 143ko)

Be aware that separators between multiple-value options are now ":" (and no more ","). For example, we now write *-l 0:1*, instead of *-l 0,1*. Please also note that the order of values for argument *-T*, *-t*, *-u* and *-U* (i.e. all options regarding the specification of margins within the Pearson' system) have been changed. For example, we now write *-T 3:10:0:3:1:0* instead of *-T 0:1:3:10:0:3* (the first 2 values becomes the last two but permuted). All the site has been updated.

Add Tcl-Tk GUIs for HMC-IN programs in order to generate automatically command line to run programs (GUIs for PMC programs are planned for the next release). Default values for options are now read in two text files called *InitHMC-IN.txt* and *InitPMC.txt*. Changing values in these files will automatically change default value in C++ and Tcl-Tk programs. Be careful when playing with these files!

Introduce *Plplot* for better visualization. If you do not want to use *Plplot* (and continue to use *Gnuplot*), comment the four lines dedicated to *Plplot* in the Makefile before compilation. All figures in the site have been updated.

The option *-f* do no exist anymore and is replaced by options *-X* and *-Y* for providing chain (X) and observations (Y) files. These options are used to provide a name file to read or to write (result of processing). All tutorials and examples in the site have been updated.

[HMC Ext 20100512.tgz](#) (version 0.3 - 122ko)

Add two programs for the conversion of a pgm image into a text file (and conversely) in order for all estimation programs to be used to segment images (see Tool section in [Help](#) menu). The 1D decomposition is based on the Hilbert-Peano scan (which has been extended to deal with even dimensions, and not only 2^n dimensions) [A15].

Add a program called *estimHMC-IN_BL* which estimates (EM/SEM/ICE) and restore (MPM and MAP) data based on sub-sampling (bootstrap) and a local estimation of forward and backward probabilities. This method is of particular interest when one wants to segment huge data sets (e.g. an image with dimensions 3000x4000). Indeed, classical method (as the one implemented in *estimHMC-IN*) can not manage such sets of data, due to

Beyond HMC

heavy memory requirements and computing demands. See [A16] for detailed explanations. [Tutorial #3](#) illustrates the program use for segmenting an image.

All estimation programs (i.e. *estimHMC-IN*, *estimHMC-IN_BL* and *estimPMC*) can now be used with a number of classes greater than 2.

[HMC_Ext_20100130a.tgz](#) (version 0.2a - 100ko)

Version a: improve Gnuplot graphics for better visualization.

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, [A17]) and margins (from the Pearson' system [A11]).

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 deposit. 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].

Installation

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

GNU Scientific Library: furnishes a quite complete collection of numerical algorithms, from [GSL library](#) random number generation to special function evaluation (and many more). Install packages *libgsl0-dev* and *libgsl0ldbl* (using synaptic or *apt-get install ...* with root privileges).

Plplot (optional) Plplot is a cross-platform software package for creating scientific plots. This program is used to generate *.eps* files from results of simulation/restoration. If you do not want to use Plplot, comment section regarding Plplot in the Makefile's preamble before compiling.

Programs have been developed in C++, and compile successfully with *gcc version 4.3.2* (under Linux/Debian 4.3.2-1.1 and Mac 10.6. You also need *make* for compilation). 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. If you successfully install programs on specific platforms, please tell me.

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

Beyond HMC

deleted and the entire set downloaded again, to ensure compatibility. After downloading, decompress and untar the functions:

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

A directory called *HMC_Ext* will be created, in which you can find a *Makefile* and 6 directories:

<i>Sources</i>	Source files (<i>.c</i> and <i>.cc</i> 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 (<i>.h</i> files) for classes. Each header file corresponds to only one class with the same name, and one <i>.cc</i> also with the same name.
<i>Programs</i>	Source files (<i>.cc</i> 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 purposes and 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>	Will contain 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.

Program compilation (it can take a minute)

```
>> make -f MakefileDist programs
ar: creating archive ./Bin/libHMC_Ext.a
Library ./Bin/libHMC_Ext.a updated.
...
restoreHMC-IN done.
simulHMC-IN done.
...
```

Test installation

To test correctness of your installation, 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*.

It is important to note that default values can be changed either by providing an initialisation file using *./simulHMC-IN -i InitFile.txt* (see *Tcl* directory for an example) or manually by specifying values for arguments such as *./simulHMC-IN -v 2 -Y essai.txt*. See [Help](#) section for a complete explanation on argument values. All options not specified will take default values.

To restore data using both MPM and MAP Bayesian criterion, type command

```
>> ./restoreHMC-IN -Y Y_HMC-IN.txt -F X_HMC-IN.txt
5.3
5.5
```

Beyond HMC

Argument for option *-Y* 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 (this second argument is not required ; in that case no error rate will be computed). The restored files are respectively saved under names *Y_HMC-IN_Rest_MPM_HMC-IN.txt* and *Y_HMC-IN_Rest_MAP_HMC-IN.txt* .

The two values show the MPM and MAP classification error rates. You can expect to get different values than those printed due to different stochastic simulation. Default parameters for restoration are the same that default parameters for simulation which justify these low error rates.

For more versatile programs, add option *-v 1*

```
>> ./simulHMC-IN -v 1

Simulation of a noisy Markov chain with independent noise

*****
./simulHMC-IN [arguments] :
  -Chain length : 1000
  -Markov matrix:
0.45 0.05
0.05 0.45

  -Markov chain file name (X): X_HMC-IN.txt
  -Observations file name (Y): Y_HMC-IN.txt
  -Laws:
    - Gaussian      (8); mu1->mu4= +0.00,+1.00,+0.00,+3.00, beta1,2= +0.00 & +3.00
    - Gaussian      (8); mu1->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 : 464, i.e. 46.4%

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

End of - Simulation of a noisy Markov chain with independent noise
```

and

```
>> ./restoreHMC-IN -Y Y_HMC-IN.txt -F X_HMC-IN.txt -v 1

Restoration of a noisy Markov chain with independent noise

*****
./restoreHMC-IN [arguments] :
  -Markov matrix:
0.45 0.05
0.05 0.45

  -Observation file name (Y) : Y_HMC-IN.txt
  -Supplied Markov chain file: X_HMC-IN.txt
  -Laws:
    - Gaussian      (8); mu1->mu4= +0.00,+1.00,+0.00,+3.00, beta1,2= +0.00 & +3.00
    - Gaussian      (8); mu1->mu4= +2.00,+1.00,+0.00,+3.00, beta1,2= +0.00 & +3.00
  -Verbose level   : 1
*****

The error rate for MPM is 5.6
The error rate for MAP is 7
```

Beyond HMC

```
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 with 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 [options]
-n Chain lenght [>50]; Actual value=1000
-Y Filename for simulated observations Y; Actual value=Y_HMC-IN.txt
-X Filename for simulated Markov chain X; Actual value=X_HMC-IN.txt
-m Joint Markov probabilities; Actual value=0.45:0.05:0.05:0.45
-T Parameters for law #1-> mean:var>0:beta1>=0:beta2>=1:sign(+1,-1); law in [0=Gaussian; 1=G
-U Parameters for law #1-> mean:var>0:beta1>=0:beta2>=1:sign(+1,-1); law in [0=Gaussian; 1=G
-x Save graphic files to current directory
-F Provide the 0/1 text file to be noisied => -m not required
-v Verbose [0/1/2]; Actual value=0
-h This help!
```

To get explanation on argument values for each program, see the lists reported in [Help](#) menu.

Help

Remark: Here, I only give some basic instructions regarding use of programs, and do 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 `-Y`. 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 make basic simulations.

There is 5 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 and restoration](#) programs, to restore a noisy signal from model parameter values estimated by using iterative routine (most of the time ICE *Iterated Conditional Estimation*, EM and SEM are available for estimation), and
- [Tool](#) programs, to compute an error rate between two binary-valued files, or to get a graphic of a density from the Pearson' system of distributions or from a specific copula...

Remark 1: All simulation and restoration programs are limited to 2 classes (binary chain). The only reason is the ease of use. Estimation programs have no such limitation. They can be used whatever the number of classes.

Remark 2: Program options are presented with their default value (hard-coded), which have been chosen to be the simplest as possible (e.g. always Gaussian margins and Gaussian copulas are chosen by default). Do not hesitate to modify the default choices by changing init value files or by providing specific values

manually.

Simulation programs

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

Simulation of data following an i.i.d. model

```
>> ./simulBLIND -n 1000 -A 0.4:0.6 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -X X_BLIND.txt -Y Y_BLIND.txt -v 0
```

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

Simulation of data following an HMC-IN model

```
>> ./simulHMC-IN -n 1000 -m 0.45:0.05:0.05:0.45 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -X X_HMC-IN.txt -Y 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 0.45:0.05:0.05:0.45 -c 1:1:1:1 -T 0:1:0:3:1:0 -t 0.3:1:0:3:1:0 -u 1.7:1:0:3:1:0 -U 2:1:0:3:1:0 -X X_PMC.txt -Y Y_PMC.txt -v 0
```

```
>> ./simulPMC -n 1000 -m 0.45:0.05:0.05:0.45 -c 1:1:1:1 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -p 1 -X X_PMC.txt -Y Y_PMC.txt -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 `./simulPMC -h`. 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. Instead, you can provide the Kendall's tau using option `-k` (e.g. `-k 0.1:0.4:0.4:0.7`). Kendall's tau belongs to $[-1,1]$ (such as correlation coefficient) but this range is not valid for all copulas (`./simulPMC -h` provides the correct range for each copula, [A17]).

Three other options are available: `-F`, `-x` and `-h`. Option `-F` is used to specify the binary chain to be noised (so that only observation are simulated).

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 i.i.d. model from provided parameters

```
>> ./restoreBLIND -A 0.4:0.5 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -Y Y_BLIND.txt -v 0
```

Four other options are available: `-F`, `-x`, `-h` and `-X`. Option `-X` is used to provide the base name of the MAP restored chain. If it is not provided, the name of the input file is used for the output files base name. Option `-F` allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Restoration using an HMC-IN model from provided parameters

```
>> ./restoreHMC-IN -m 0.45:0.05:0.05:0.45 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -Y Y_HMC-IN.txt -v 0
```

Four other options are available: `-F`, `-x`, `-h` and `-X`. Option `-X` is used to provide the base name of the MAP and MPM restored chain. If it is not provided, the name of the input file is used for the output files base name. Option `-F` allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Restoration using a PMC model from provided parameters

```
>> ./restorePMC -m 0.45:0.05:0.05:0.45 -c 1:1:1:1 -T 0:1:0:3:1:0 -t 0.3:1:0:3:1:0 -u 1.7:1:0:3:1:0 -U 2:1:0:3:1:0 -Y Y_PMC.txt -v 0
```

```
>> ./restorePMC -m 0.45:0.05:0.05:0.45 -c 1:1:1:1 -T 0:1:0:3:1:0 -U 2:1:0:3:1:0 -Y Y_PMC.txt -p 1 -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 `-h`. 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.

Four other options are available: `-F`, `-x`, `-h` and `-X`. Option `-X` is used to provide the base name of the MAP and MPM restored chain. If it is not provided, the name of the input file is used for the output files base name. Option `-F` allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

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).

Remark Programs `estimBLIND` and `estimHMC-IN` can handle several input files, using for example `-Y R.txt:G.txt:B.txt`. Of course, all files must be the same length. The multivariate data driven densities (whose dimension is given by the number of input files) can be choosen in the set of available copulas (a list is given

using -h).

Estimation and restoration using an i.i.d. model

```
>> ./estimBLIND -E EM -e 300 -K 2 -l 0:0 -Y Y_BLIND.txt -v 0
```

Four other options are available: -F, -x, -h and -X. Option -X is used to provide the base name of the MAP and MPM restored chain. If it is not provided, the name of the input file is used for the output files base name. Option -F allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Estimation and restoration using an HMC-IN model

```
>> ./estimHMC-IN -E EM -e 50 -K 2 -l 0:0 -Y Y_HMC-IN.txt -v 0
```

Four other options are available: -F, -x, -h and -X. Option -X is used to provide the base name of the MAP restored chain. If it is not provided, the name of the input file is used for the output files base name. Option -F allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Estimation and restoration using an HMC-IN model (sub-sampling estimation)

```
>> ./estimHMC-IN_BL -E EM -e 50 -K 2 -l 0:0 -s 5 -b 3 -P 0.2 -Y Y_HMC-IN.txt -v 0
```

Option -s is used to define the width of the 1D window around the pixel of interest. Option -b defines the way sub-sampling is performed. Option -P provides the percent of data to be used when option b is set to 1 or 2 (for option 3, the percent is computed automatically). Note that the default estimation method is EM here. Indeed, SEM and ICE require the simulation of a posteriori chain which can only be achieved by sub-optimal realization, called "marginal realization" (according to the method described in [A2]).

Four other options are available: -F, -x, -h and -X. Option -X is used to provide the base name of the MAP and MPM restored chain. If it is not provided, the name of the input file is used for the output files base name. Option -F allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Estimation and restoration using a PMC model

```
>> ./estimPMC -E ICE -e 50 -K 2 -c 1:1:1:1 -l 0:0:0:0 -Y Y_PMC.txt -v 0
```

```
>> ./estimPMC -E ICE -e 50 -K 2 -c 1:1:1:1 -l 0:0:0:0 -Y Y_PMC.txt -p 1 -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, options -l and -c always take a list of K integers (K in the number of classes set with option -K), whereas the true PMC model requires K^2 integers.

A list of available copulas can be printed using option -h ([A17])

Four other options are available: -F, -x, -h and -X. Option -X is used to provide the base name of the MAP and MPM restored chain. If it is not provided, the name of the input file is used for the output files base name. Option -F allows to specify the original binary file in order for the program to compute the error rate between this file and the result of restoration.

Tool programs

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

How to convert a pgm image into a text file for latter use?

```
>> ./pgm2Txt -Y image.pgm -X image.txt -v 0
```

The image is decomposed by using the Peano scan [A15]. The name of the generated text file is *image.pgm.txt*.

One other option is available: -h.

Conversely, how to convert a txt file into a pgm image (reconstruction after segmentation)?

```
>> ./txt2Pgm -Y image.txt -X imageRec.pgm -D line:col -v 0
```

The [Tutorial #3](#) shows how to use this two programs. The image is reconstructed by using the inverse Peano scan. The name of the generated pgm image is *image.pgm.txt.pgm*.

One other option is available: -h.

Error rate between two binary files

```
>> ./errorRate -i X_PMC.txt:Y_PMC_ICE_MAP_PMC.txt -v 0
```

One other option is available: -h.

Draw one or two densities from the Pearson system of distributions

```
>> ./drawPearsonPdf -T -10:10:0:4:1:6 -U 5:20:1.5:4:-1:1 -v 0 -x
```

One other option is available: -h. By default, option -x save graphics to the current directory.

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

```
>> ./drawCopula2D -c 3 -T 11:6:0.7:4.3:1:6 -U 10:5:3:9.5:1:6 -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 save graphics to current directory. To avoid question about parameters during execution time, you can specify Kendall's Tau using option -k.

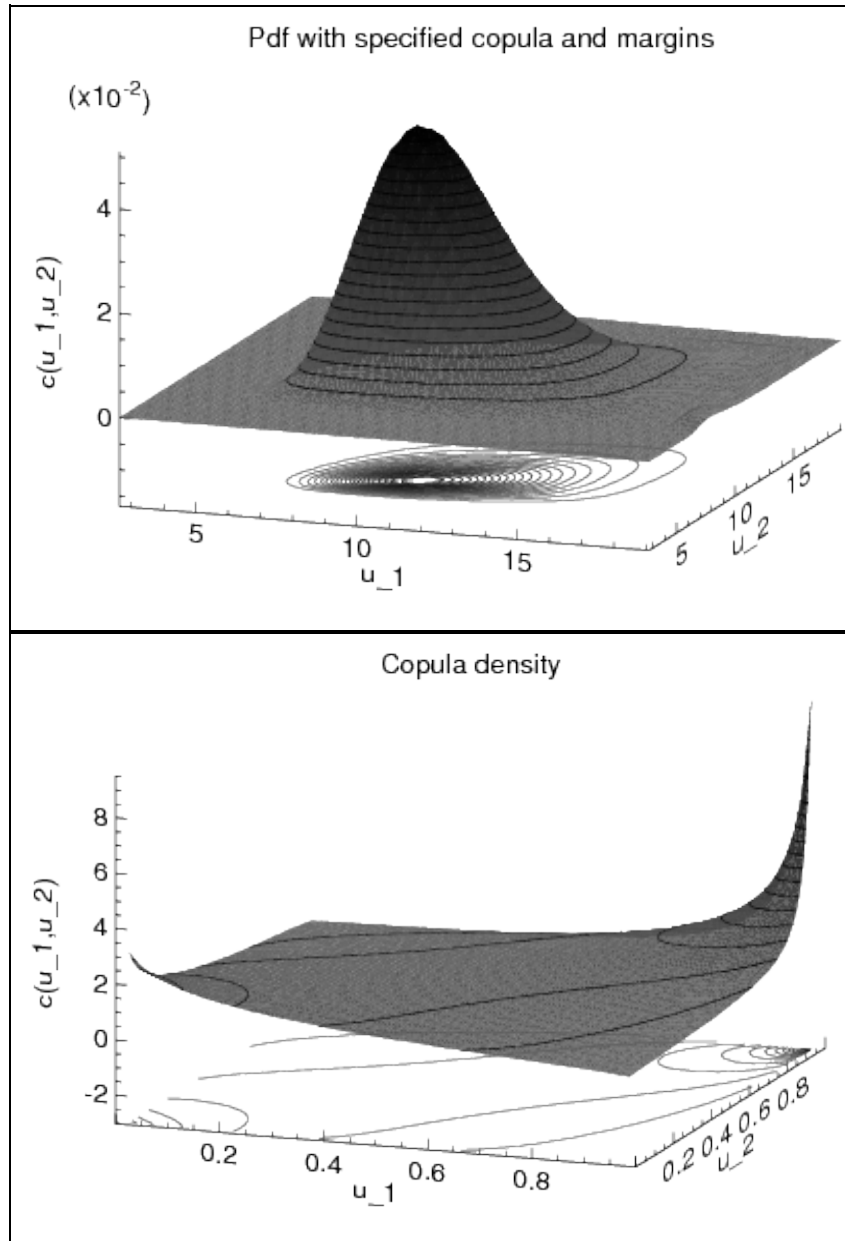
The example above gives the following display:

```
Draw of a 2D pdf with specified copula and margins
*****
drawCopula2D [arguments] :
  -Margin laws          :
    - Beta 2           (6); mu1->mu4= 11.00,6.00,12.30,154.80, beta1,2= 0.70 & 4.30
    - Type V          (5); mu1->mu4= 10.00,5.00,19.36,237.50, beta1,2= 3.00 & 9.50
```

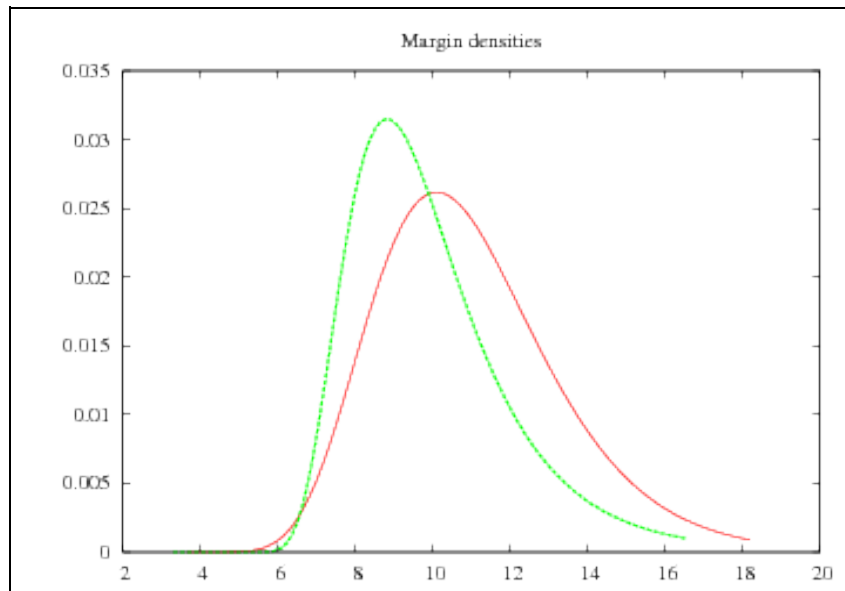
Beyond HMC

```
-Copula type           : 3 (tau=0.47, theta=1.89)
-Verbose level        : 1
*****
End of - Draw of a 2D pdf with specified copula and margins
```

and generates the following figures (using PIPLOT):



Beyond HMC



Another option is available: `-h`.

Product Copula

- 1 -> Gaussian Copula (Kendall Tau in [-1; 1])
- 2 -> Student Copula (Kendall Tau in [-1; 1])
- 3 -> Gumbel-Hougaard Copula (Kendall Tau in [0; 1])
- 4 -> Farlie-Gumbel-Morgenstern Copula (Kendall Tau in [-2/9; 2/9])
- 5 -> Cubic section Copula (Kendall Tau in [0; 33/200])
- 6 -> Clayton Copula (Kendall Tau in [0; 1])
- 7 -> A12 Copula (Kendall Tau in [1/3; 1])
- 8 -> A14 Copula (Kendall Tau in [1/3; 1])

Be aware that available copulas can be different from one program to the other. Use option `-h` to get a list of available copulas for the program of interest ([A17]). Of course, you can mix copulas, e.g. `-c 0:1:2:3`, but expect to get very strange results!

`-k` list of reals

no default value

To set the list of Kendall's tau for each copula required by the program. For example `-k 0.4:0.6:0.7:0.2` in conjunction with `-c 1,1,1,1` means that you define 4 Gaussian copulas with different correlation coefficients. All tau values must be between [-1,1] but some copulas restrict the range (see above). To know the range for each copula, use option `-h`.

`-x` string

no default value

Set the repositories where `.eps` 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.

-->

Tutorial

Here is a collection of, simple or more sophisticated, examples to illustrate possible uses of programs. This list will grow 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 PIPlot during execution using `-x` option (assuming that programs have been compiled with PIPlot 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, which is itself strictly more general than BLIND (i.e. i.i.d.) model".
 - [Tutorial #3](#): How to segment a huge image with a bootstrap parameters estimation for the HMC model?
-

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

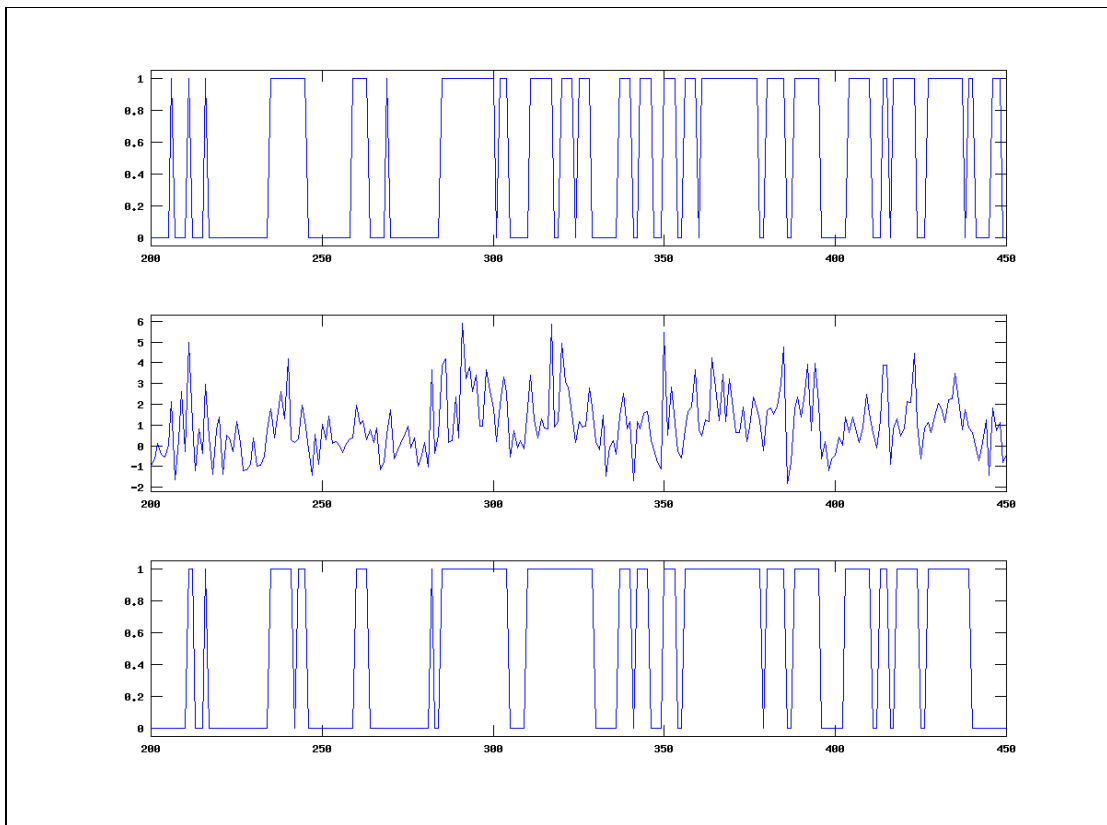
Simulation of an HMC-IN mixture of two gamma laws

```
>> ./simulHMC-IN -n 10000 -m 0.4:0.1:0.1:0.4 -T 0:1:1:4.5:1:1 -U 2:2:2:6:1:1 -l 1:1 -X X_HMC-IN.txt -Y
Y_HMC-IN.txt -v 1
and estimation using 100 ICE iterations
```

```
>> ./estimHMC-IN -F X_HMC-IN.txt -E EM -e 100 -l 1:1 -v 1
```

This gives MPM and MAP error rates of 11.07% and 11.51% respectively. Next figure shows the simulated Markov chain (up), the simulated observed data (middle) and the ICE estimation and MPM restoration (down).

Beyond HMC



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).

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

```
- Gamma      (3); mu1->mu4= -0.01,0.99,0.94,4.27, beta1,2= 0.91 & 4.37  
- Gamma      (3); mu1->mu4=  2.04,2.03,3.79,22.97, beta1,2= 1.73 & 5.60
```

whereas Markov joint a priori matrix is given by

```
0.40  0.09  
0.09  0.41
```

with a priori probabilities:

```
0.49  
0.51
```

and transition matrix:

```
0.81  0.19
```

0.19 0.81

Estimated parameters are close to the parameters used for simulation.

Tutorial #2: "PMC model is strictly more general than HMC-CN, which is itself strictly more general than HMC-IN, which is itself strictly more general than BLIND (i.e. i.i.d.) model"

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:

```
- Joint a priori matrix :
    0.45    0.05
    0.05    0.45
- Laws
      **** (0,0) ****
      ->Copula : Gaussian - tau = 0.50 -
    1.00    0.71
    0.71    1.00
      ->Margins
      - Cl 0, Law: Gaussian    (8); Mu = 0.00, Sigma = 1.00
      - Cl 0, Law: Gaussian    (8); Mu = 0.00, Sigma = 1.00
      **** (0,1) ****
      ->Copula : Gaussian - tau = 0.50 -
    1.00    0.71
    0.71    1.00
      ->Margins
      - Cl 1, Law: Gaussian    (8); Mu = 0.30, Sigma = 1.00
      - Cl 2, Law: Gaussian    (8); Mu = 1.70, Sigma = 1.00
      **** (1,0) ****
      ->Copula : Gaussian - tau = 0.50 -
    1.00    0.71
    0.71    1.00
      ->Margins
      - Cl 2, Law: Gaussian    (8); Mu = 1.70, Sigma = 1.00
      - Cl 1, Law: Gaussian    (8); Mu = 0.30, Sigma = 1.00
      **** (1,1) ****
      ->Copula : Gaussian - tau = 0.50 -
    1.00    0.71
    0.71    1.00
      ->Margins
      - Cl 3, Law: Gaussian    (8); Mu = 2.00, Sigma = 1.00
      - Cl 3, Law: Gaussian    (8); Mu = 2.00, Sigma = 1.00
```

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):

Tutorial #2: "PMC model is strictly more general than HMC-CN, which is itself strictly more general than HMC-IN, which is itself strictly more general than BLIND (i.e. i.i.d.) model"

Beyond HMC

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

which gives a MPM error rate of 14.06% and a MAP-one of 15.13%.

```
>> ./restorePMC -F X_PMC.txt -p 0
```

which gives a MPM error rate of 14.75% and a MAP-one of 16.42%.

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

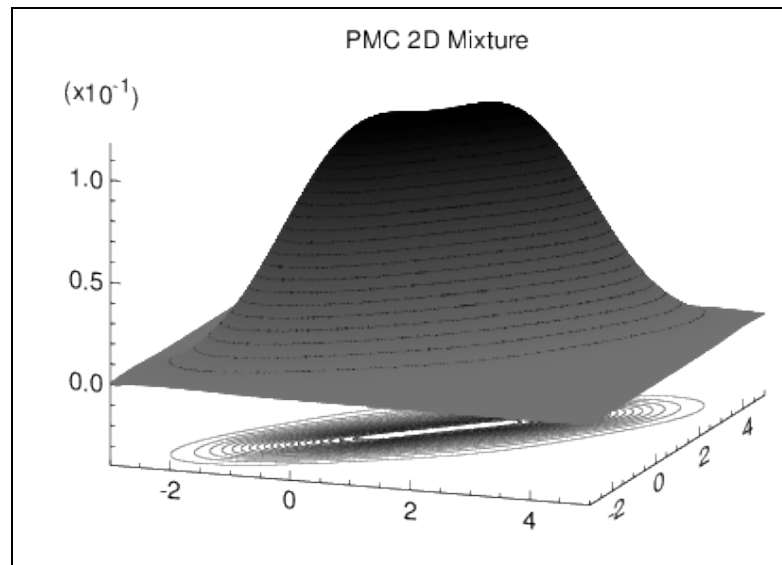
which gives a MPM error rate of 15.87% and a MAP-one of 15.86% (we must specify option -Y since, by default, the program looks for a file named *Y_HMC-IN.txt*).

```
>> ./restoreBLIND -Y Y_PMC.txt -F X_PMC.txt
```

which gives a MAP classification error rate of 16.69% (we must specify option -Y since, by default, the program looks for a file named *Y_BLIND.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 PIPlot is installed). For $(x_n=1, x_{n+1}=1)$, we get the following draw

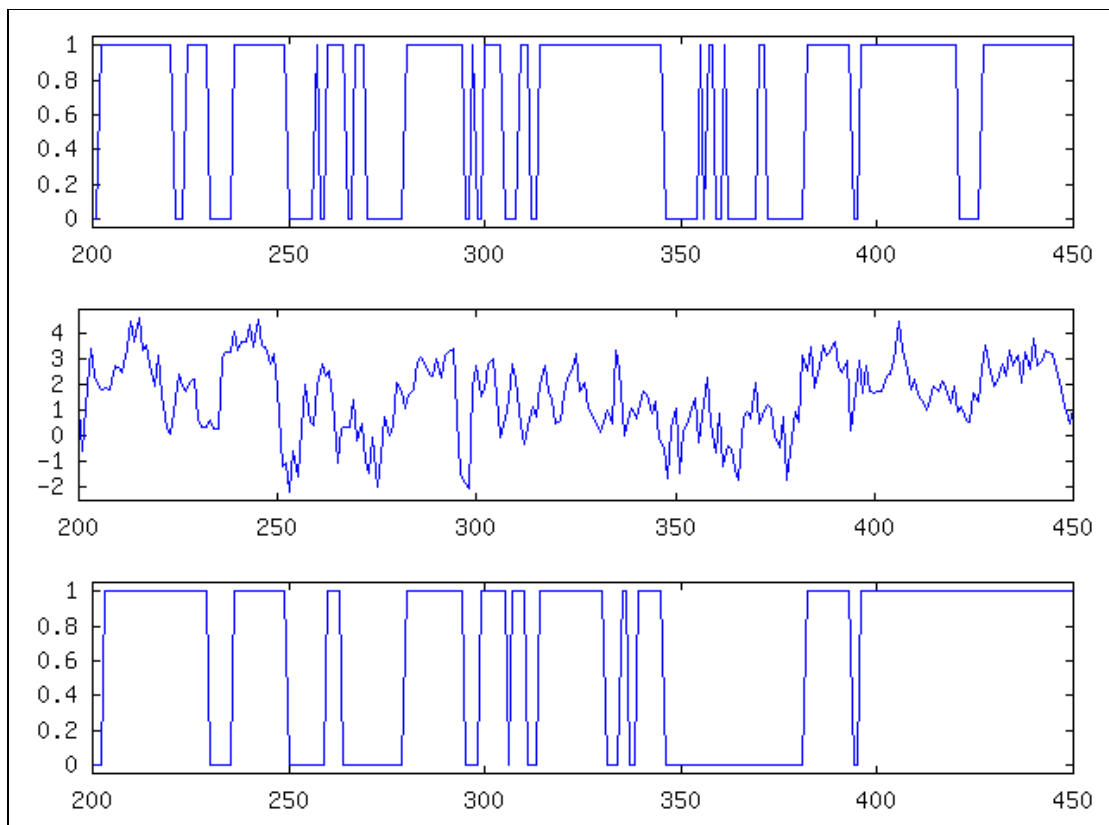


Signals can be observed with a call to octave script *Example1.m* :

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

we get

Beyond HMC



Tutorial #3: "How to segment a huge image with a bootstrap parameters estimation for the HMC model?"

Experiments are conducted with the free image [here](#). The image is sized 2000x3008. Here is a snapshot of the original image:

Beyond HMC



Original image (256 gray-levels)

The first step is to convert the pgm image into a text file using the *pgm2Txt* program:

```
>> ./pgm2Txt -Y img.pgm -X img.txt
```

This way you get a file named *img.pgm.txt*. It takes about one minute for my PC to convert the image.

Then you can segment the image according to this call (4 classes):

```
>> ./estimHMC-IN_BL -Y img.pgm.txt -K 4 -s 9 -v 1
Time spend during following functions (in sec.):
  Estimation time (EM).....348
  Restoration time for EM-MPM.....13
  Restoration time for EM-MAP.....3
```

It is worth noting that trying to use *./estimHMC-IN* or *./estimPMC* will conduct to memory overflow. According to the technique used for sub-sampling, the number of pixels that participates to parameters estimation is 1609 (i.e. 0.027%). After 50 EM iterations, you get two text files which contain the MAP and MPM segmentations: *img.pgm_EM_MPM_HMC-IN_BL.txt* and *img.pgm_EM_MAP_HMC-IN_BL.txt*.

It is then required to reconstruct the two segmentations:

```
>> ./txt2Pgm -Y img.txt_SEM_MPM_HMC-IN_BL.txt -D 2000:3008 -X
img.txt_SEM_MPM_HMC-IN_BL.pgm
>> ./txt2Pgm -Y img.txt_SEM_MAP_HMC-IN_BL.txt -D 2000:3008 -X
img.txt_SEM_MAP_HMC-IN_BL.pgm
```

Please note that images can appear black since each pixel contains the class number (0,1,2,3,...). To stretch the class numbers to visible values (between 0 and 255), you can use the Color/Auto menu in GIMP software for

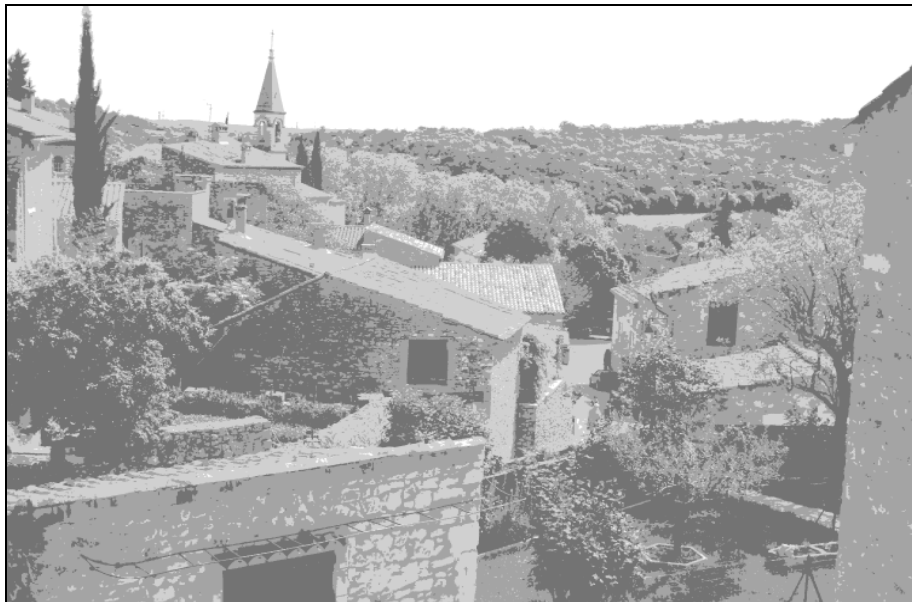
Beyond HMC

example.

Here is the results I get for MAP segmentation (dimensions of images have been reduced):



HMC (local EM estimation and MAP) segmentation (4 classes) - 50 iterations, 8 minutes



HMC (EM estimation and MAP) segmentation (4 classes) - 50 iterations, 66 minutes

Beyond HMC



Blind (EM/MAP) segmentation (4 classes) - 300 iterations, 38 minutes



Kmeans segmentation (4 classes) - 50 iterations, 21 seconds

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.

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

- [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](#)
 - [A15] W. Skarbek, *Generalized Hilbert scan in image printing*, Theoretical Foundations of Computer Vision, p. 45-57, R. Klette et W. G. Kropetsh Ed., Akademik Verlag, 1992.
 - [A16] S. Derrode, L. Benyoussef and W. Pieczynski, *Sub-sample-based HMC estimation with application to large data sets segmentation*, submitted, 2010.
 - [A17] S. Derrode *Some details about a few copulas*, Research Report, Ecole Centrale Marseille & Institut Fresnel, February 2011. [Link](#)
 - [A18] S. Derrode, W. Pieczynski, *Unsupervised restoration in Gaussian pairwise mixture models*, Eusipco'11, Barcelona (Spain), August 29 - September 2, 2011.
 - [A19] S. Derrode, W. Pieczynski, *Segmentation d'images par modèle de mélange conjoint non gaussien*, submitted to Traitement du Signal in September 2011.
-

Triplet Markov Chain, 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](#)
- [B5] N. Brunel, J. Lapuyade-Lahorgue and W. Pieczynski, *Modeling and unsupervised classification of multivariate hidden Markov chains with copulas.*, IEEE Trans. on Automatic Control, Vol. 55(2), pp. 338-349, 2010. [Link](#)
- [B6] P. Lanchantin, J. Lapuyade-Lahorgue and W. Pieczynski, *Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise.*, Signal Processing, Vol. 91, pp. 163-175, 2011. [Link](#)
- [B7] W. Pieczynski, *Triplet Markov chain and image segmentation*, chapter 4 in *Inverse problems in Vision and 3D Tomography*, Ed. A. Mohammed-Djafari, Wiley, 2010. [Link](#)

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

To do list (short term) - before publishing next release (version: 0.8)

- Include multi-scale mixtures model (see ref [A18, A19])
- Include multi-scale HMC-IN model

To do list (middle term) - after the next release (for version: 0.9)

- Implement the Triplet Markov chain model (see ref. [B6]) and include copulas. Add a tutorial
- Implement long Memory HMC model (as a TMC model)

To do list (long term)

- Fuzzy HMC model
- Higher-order HMC model