

OpenBDLM: an open-source Matlab software for
time-series analysis using Bayesian dynamic
linear models

Ianis Gaudot, Luong H. Nguyen, James-A. Goulet
Polytechnique Montreal

October 24, 2018

Contents

1	What OpenBDLM does ?	7
2	Installing OpenBDLM	9
2.1	Prerequisites	9
2.2	Installing	9
2.3	Getting started	9
2.4	Demo	10
3	Bayesian dynamic linear models	11
3.1	Transition equation	11
3.2	Observation equation	11
4	Data processing	13
4.1	Purpose	13
4.2	Input data format	13
4.2.1	CSV files	13
4.2.2	.MAT files	14
4.3	Output data format	14
4.4	Merging timestamps vector	14
4.5	Data processing functions	14
5	Model building	15
5.1	Purpose	15
5.2	Model class	15
5.3	Dependencies	15
5.3.1	Observed covariate	15
5.3.2	Hidden covariate	15
5.4	Model components	15
5.4.1	Local level	15
5.4.2	Local trend	15
5.4.3	Local acceleration	15
5.4.4	Local level compatible trend	15
5.4.5	Local level compatible acceleration	15
5.4.6	Local trend compatible acceleration	15

5.4.7	Periodic	15
5.4.8	Kernel regression	15
5.4.9	Residual - first order autoregressive	15
5.5	Model building functions	15
6	Model parameters learning	17
6.1	Purpose	17
6.2	Posterior	17
6.2.1	Prior	17
6.2.2	Likelihood	17
6.3	Model parameters bounds and transformed spaces	17
6.3.1	Logarithmic transformation	17
6.3.2	Sigmoid transformation	17
6.4	Gradient-based optimization techniques	17
6.4.1	Newton-Raphson approach	17
6.4.2	Stochastic Gradient Ascent approach	17
6.5	Constrain model parameters between each others	17
6.6	Model parameters learning functions	17
7	Hidden states estimation	19
7.1	Purpose	19
7.2	Kalman equations	19
7.3	UD computations	19
7.4	Filtering	19
7.5	Smoothing	19
7.6	Hidden states estimation functions	19
8	Data simulation	21
8.1	Purpose	21
8.2	Data simulation functions	21
9	Visualization tools	23
9.1	Purpose	23
9.2	Data availability plots	23
9.3	Hidden states plots	23
9.4	Export figures options	23
9.5	Visualization tools functions	23
10	Version control	25
10.1	Purpose	25
10.2	Version control functions	25
11	OpenBLDM options	27

<i>CONTENTS</i>	5
12 Examples	29
12.1 Example 1	29
12.2 Example 2	29
12.3 Example 3	29
12.4 Example 4	29
13 List of functions	31
14 Older versions	33
15 Last changes	35

Chapter 1

What OpenBDLM does ?

OpenBDLM is a Matlab open-source software designed to use Bayesian Dynamic Linear Models for long-term time series analysis (i.e time step of in the order of one hour or higher). OpenBDLM is capable to process simultaneously any time series data to interpret, monitor and predict their long-term behavior. OpenBDLM also includes an anomaly detection tool which allows to detect abnormal behavior in a fully probabilistic framework.

- linear gaussian state-space models
- time-series decomposition
- anomaly detection
- model identification
- data visualization

Chapter 2

Installing OpenBDLM

These instructions will get you a copy of the project up and running on your local machine for direct use, testing and development purposes.

2.1 Prerequisites

Matlab (version 2016a or higher) installed on Mac OSX or Windows

2.2 Installing

1. Remove from your Matlab path all previously OpenBDLM versions
2. Extract the ZIP file (or clone the git repository) somewhere you can easily reach it.
3. Add **OpenBDLM-master/** folder and all the sub folders to your path in Matlab :
 - using the “Set Path” dialog box in Matlab, or
 - by running `addpath` function from the Matlab command window

2.3 Getting started

Enter in the folder **OpenBDLM-master**, and type `OpenBDLM.main();` in the Matlab command line. The OpenBDLM starting menu should appear on the Matlab command window (2.1).

Listing 2.1: OpenBDLM starting menu



```

Starting OpenBDLM_V???...

Structural Health Monitoring using Bayesian Dynamic Linear Models

- Start a new project:

    *      Enter a configuration filename
    0  -> Interactive tool

- Type D to Delete project(s), V for Version control, Q to Quit.

choice >>

```

2.4 Demo

In the Matlab command line, type `run_DEMO.m` to run a little demo. You should see some messages on the Matlab command window showing that the programs runs properly:

Listing 2.2: Output on Matlab command window when running `run_DEMO.m` script

```

Starting OpenBDLM_V???...
Starting a new project...
Building model...
Simulating data...
Plotting data...
Saving database (binary format) ...
Saving database (csv format) ...
Saving project...
Printing configuration file...
Saving database (binary format) ...
Saving project...
See you soon !

```

If you do not see anything except Matlab errors verify your Matlab version, and your Matlab path. Be sure that you run the program from the top-level of OpenBDLM-master folder, not from another folder.

Chapter 3

Bayesian dynamic linear models

Bayesian dynamic linear models (hereafter referred to as BDLMs) are a class of linear gaussian state-space models which can be described from the transition and the observation equations.

3.1 Transition equation

The transition equation describes the dynamics of the system, and is formulated as

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{w}_t, \quad \begin{cases} \mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \\ \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t), \end{cases} \quad (3.1)$$

where, for each time $t = 1, \dots, T$, the variables \mathbf{x}_t follow a Gaussian distribution with mean $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$, \mathbf{A}_t is the transition matrix, and \mathbf{w}_t represents Gaussian model errors with zero mean and covariance matrix \mathbf{Q}_t . The variables \mathbf{x}_t are usually referred to as hidden states because they are not directly observed.

3.2 Observation equation

The relationship between the observations \mathbf{y}_t and the hidden states \mathbf{x}_t is given by the observation equation, such as

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{x}_t + \mathbf{v}_t, \quad \begin{cases} \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t), \end{cases} \quad (3.2)$$

where \mathbf{C}_t is the observation matrix, and \mathbf{v}_t is the Gaussian measurement error with zero mean and covariance matrix \mathbf{R}_t . BDLMs are capable to analyze multiple time series simultaneously. In case of dependencies between the time series,

regression coefficients are added in \mathbf{C}_t . One particularity of BDLMs is their capacity to update the current predicted state with the current observations using the Kalman filter equations. The Kalman filter equations are computationnally efficient, as they are solved fully analytically. BDLMs are capable to process non-stationnary time series without retraining the model.

Chapter 4

Data processing

4.1 Purpose

As mentioned earlier, BDLMs are capable to analyze simultaneously multiple time-series. Multiple time-series analysis is particularly useful to incorporate information from observed environmental effects by creating dependencies between time-series. However, in most cases, the set of available time-series is heterogeneous, in the sense that each time-series do not originate from the same system of acquisition. Therefore, the raw data do not usually share the same timestamps. This is an issue because BDLM techniques are not capable to analyze asynchronous time-series. The data pre-processing in OpenBDLM is thus dedicated to synchronize the time-series between each others. The preprocessing in OpenBDLM is minimalist and focuses on time synchronization. Neither outlier removal nor normalization is done to preserve the genuine information from the data.

4.2 Input data format

OpenBDLM accept two types of data formatting depending whether the time-series are already synchronized, or not.

4.2.1 CSV files

Comma Separated Values (CSV) data formatting is generally used to load time-series data, which have not been synchronized yet. The user should provide one CSV files for each time series. The file is a two columns file that should be organized as follow:

Listing 4.1: CSV file example

```
'name' , '2000-01-01-22-00-00'
```

```
737422      ,      0.40
737423      ,      0.21
737424      ,      0.548
```

The first line of the file is the header. In the header, the first field should contain the label of the time-series given as a quoted delimited string, as 'name'. The second field should be the date of the first timestamp given as a quoted delimited string, as 'YYYY-DD-MM-HH-MM-SS'. Note that the two fields in the header are not the label of each column. The others line have two fields. The first field is the date given as a serial date number. The second field is the magnitude of the physical quantity measured (the measurement), given as a real number.

4.2.2 .MAT files

4.3 Output data format

4.4 Merging timestamps vector

4.5 Data processing functions

Chapter 5

Model building

5.1 Purpose

5.2 Model class

5.3 Dependencies

5.3.1 Observed covariate

5.3.2 Hidden covariate

5.4 Model components

5.4.1 Local level

5.4.2 Local trend

5.4.3 Local acceleration

5.4.4 Local level compatible trend

5.4.5 Local level compatible acceleration

5.4.6 Local trend compatible acceleration

5.4.7 Periodic

5.4.8 Kernel regression

5.4.9 Residual - first order autoregressive

5.5 Model building functions

Chapter 6

Model parameters learning

6.1 Purpose

6.2 Posterior

6.2.1 Prior

6.2.2 Likelihood

6.3 Model parameters bounds and transformed spaces

6.3.1 Logarithmic transformation

6.3.2 Sigmoid transformation

6.4 Gradient-based optimization techniques

6.4.1 Newton-Raphson approach

6.4.2 Stochastic Gradient Ascent approach

6.5 Constrain model parameters between each others

6.6 Model parameters learning functions

Chapter 7

Hidden states estimation

7.1 Purpose

7.2 Kalman equations

7.3 UD computations

7.4 Filtering

7.5 Smoothing

7.6 Hidden states estimation functions

Chapter 8

Data simulation

8.1 Purpose

8.2 Data simulation functions

Chapter 9

Visualization tools

9.1 Purpose

9.2 Data availability plots

9.3 Hidden states plots

9.4 Export figures options

9.5 Visualization tools functions

Chapter 10

Version control

10.1 Purpose

10.2 Version control functions

Chapter 11

OpenBLDM options

Chapter 12

Examples

12.1 Example 1

Simulated data with: one time series, one model class, $\{[12\ 31\ 41]\}$.

12.2 Example 2

Simulated data with: two time series with dependencies, one model class, $\{[11\ 41], [11\ 31\ 41]\}$.

12.3 Example 3

Simulated data with one time series, two model classes, $\{[21\ 31\ 41]\}$ and $\{[12\ 31\ 41]\}$.

12.4 Example 4

Real data with one time series, one model class, $\{[12\ 51\ 41]\}$.

Chapter 13

List of functions

Chapter 14

Older versions

Chapter 15

Last changes