

## Hidden State Inference

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	/Users/gilesc/src/HMM-MAR/hidden_state_inference/hidden_state_inference.cpp File Reference .	3
2.1.1	Detailed Description . . . . .	3
2.1.2	Function Documentation . . . . .	4
2.1.2.1	hidden_state_inference(arma::mat &gamma, arma::mat &Xi, arma::colvec &scale, const arma::mat &B, const arma::rowvec &Pi_0, const arma::mat &P, const int order) . . . . .	4
2.2	/Users/gilesc/src/HMM-MAR/hidden_state_inference/hidden_state_inference_mx.cpp File Reference	4
2.2.1	Detailed Description . . . . .	5
2.2.2	Function Documentation . . . . .	5
2.2.2.1	mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]) . . . . .	5
2.3	hidden_state_inference_test.cpp File Reference . . . . .	5
2.3.1	Detailed Description . . . . .	6
2.3.2	Function Documentation . . . . .	6
2.3.2.1	main(int argc, char *const argv[]) . . . . .	6
	<b>Index</b>	<b>7</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">/Users/gilesc/src/HMM-MAR/hidden_state_inference/hidden_state_inference.cpp</a>	
Hidden state inference for the HMM . . . . .	3
/Users/gilesc/src/HMM-MAR/hidden_state_inference/ <b>hidden_state_inference.hpp</b> . . . . .	??
/Users/gilesc/src/HMM-MAR/hidden_state_inference/ <a href="#">hidden_state_inference_mx.cpp</a>	
Mex interface for hidden state inference in VB-HMM . . . . .	4
/Users/gilesc/src/HMM-MAR/hidden_state_inference/ <b>hidden_state_inference_mx.hpp</b> . . . . .	??
<a href="#">hidden_state_inference_test.cpp</a>	
Test working of <a href="#">hidden_state_inference.cpp</a> . . . . .	5
<b>hidden_state_inference_test.hpp</b> . . . . .	??



## Chapter 2

# File Documentation

### 2.1 /Users/gilesc/src/HMM-MAR/hidden\_state\_inference/hidden\_state\_inference.cpp File Reference

Hidden state inference for the HMM.

```
#include "hidden_state_inference.hpp"
```

#### Functions

- void [hidden\\_state\\_inference](#) (arma::mat &gamma, arma::mat &Xi, arma::colvec &scale, const arma::mat &B, const arma::rowvec &Pi\_0, const arma::mat &P, const int order)  
*hidden state inference*

#### 2.1.1 Detailed Description

Hidden state inference for the HMM.

Hidden state inference for VB-solved HMM, using the forwards-backwards algorithm.

This code is an implementation of Diego Vidaurre's nodecluster algorithm for C++.

Giles Colclough 2 Feb 2016

Copyright 2016 OHBA

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

\$LastChangedBy\$ \$Revision\$ \$LastChangedDate\$ Contact: [giles.colclough@ohba.ox.ac.uk](mailto:giles.colclough@ohba.ox.ac.uk) Originally written on: MACI64 by Giles Colclough, 02-Feb-2016

## 2.1.2 Function Documentation

2.1.2.1 `void hidden_state_inference ( arma::mat & gamma, arma::mat & Xi, arma::colvec & scale, const arma::mat & B, const arma::rowvec & Pi_0, const arma::mat & P, const int order )`

hidden state inference

`void hidden_state_inference(gamma, Xi, B, Pi_0, P)`

A function to perform inference of hidden states in an HMM.

The outputs, *gamma* and *Xi*, need to be passed in with memory already declared.

*gamma* - (nSamples - order) x nClasses

*Xi* - (nSamples-1-order) x nClasses\*\*2

*scale* - (nSamples x 1)

The inputs are

*B* - nSamples x nClasses

*Pi\_0* - 1 x nClasses

*P* - nClasses x nClasses

*order* - integer

### Parameters

in	<i>gamma</i>	Probability of hidden state given data
in	<i>Xi</i>	Probability of hidden state given child and parent states, given data
in	<i>scale</i>	scaling factor of alpha.
in	<i>B</i>	Probability of individual data point, under observation model for each state
in	<i>Pi_0</i>	Initial state probabilities
in	<i>P</i>	State transition probabilities
in	<i>order</i>	order of MAR model [default = 0 for MVN]

## 2.2 /Users/gilesc/src/HMM-MAR/hidden\_state\_inference/hidden\_state\_inference\_mx.cpp File Reference

Mex interface for hidden state inference in VB-HMM.

```
#include "hidden_state_inference_mx.hpp"
```

### Functions

- void `mexFunction` (int nlhs, mxArray \*plhs[], int nrhs, const mxArray \*prhs[])  
*Mex interface for hidden state inference using forward-backward algorithm.*



### 2.2.1 Detailed Description

Mex interface for hidden state inference in VB-HMM.

Hidden state inference for VB-solved HMM, using the forwards-backwards algorithm.

To use this file, you may need to recompile, changing the top of the makefile to reflect your matlab installation

Copyright 2016 OHBA

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

\$LastChangedBy\$ \$Revision\$ \$LastChangedDate\$ Contact: [giles.colclough@ohba.ox.ac.uk](mailto:giles.colclough@ohba.ox.ac.uk) Originally written on: MACI64 by Giles Colclough, 14-Jul-2015

### 2.2.2 Function Documentation

2.2.2.1 void mexFunction ( int *nlhs*, mxArray \* *plhs*[], int *nrhs*, const mxArray \* *prhs*[] )

Mex interface for hidden state inference using forward-backward algorithm.

mexFunction(int num\_outputs, mxArray\* outputs, int num\_inputs, mxArray\* inputs)

HIDDEN\_STATE\_INFERENCE using forward-backward propagation

[GAMMA,XI,SCALE] = HIDDEN\_STATE\_INFERENCE(B, PI\_0, P, ORDER)

returns the probability of each state given the data, GAMMA, and the probability of each state given the parents and children, XI, together with the scalings on alpha, SCALE.

The function uses the marginal likelihood of the data given the priors for each state, the initial probabilities, PI\_0, and the transition probability matrix A. There is also an order term, relevant for MAR models. If you're not using a MAR model, set order = 0.

You MUST have four inputs and three outputs. If you don't want to make / assign variables, use ~ as a dummy variable e.g. [g,x,~] = hidden\_state\_inference\_mx(B,PI,P,ORDER)

## 2.3 hidden\_state\_inference\_test.cpp File Reference

Test working of [hidden\\_state\\_inference.cpp](#).

```
#include "hidden_state_inference_test.hpp"
```

## Functions

- `int main (int argc, char *const argv[ ])`  
*Run tests on `hidden_state_inference`.*

### 2.3.1 Detailed Description

Test working of `hidden_state_inference.cpp`.

Compile and run to test `hidden_state_inference.cpp`

Giles Colclough 2 Feb 2016

Copyright 2015 OHBA

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

\$LastChangedBy\$ \$Revision\$ \$LastChangedDate\$ Contact: [giles.colclough@ohba.ox.ac.uk](mailto:giles.colclough@ohba.ox.ac.uk) Originally written on: MACI64 by Giles Colclough, 02-Feb-2016

### 2.3.2 Function Documentation

#### 2.3.2.1 `int main ( int argc, char *const argv[ ] )`

Run tests on `hidden_state_inference`.

Performs two runs of hidden state inference, to check interface and performance.

# Index

/Users/gilesc/src/HMM-MAR/hidden\_state\_inference/hidden↔  
\_state\_inference.cpp, [3](#)

/Users/gilesc/src/HMM-MAR/hidden\_state\_inference/hidden↔  
\_state\_inference\_mx.cpp, [4](#)

hidden\_state\_inference  
hidden\_state\_inference.cpp, [4](#)

hidden\_state\_inference.cpp  
hidden\_state\_inference, [4](#)

hidden\_state\_inference\_mx.cpp  
mexFunction, [5](#)

hidden\_state\_inference\_test.cpp, [5](#)  
main, [6](#)

main  
hidden\_state\_inference\_test.cpp, [6](#)

mexFunction  
hidden\_state\_inference\_mx.cpp, [5](#)