Package ‘fHMM’

June 16, 2021

Type Package

Title Fitting Hidden Markov Models to Financial Data

Version 0.3.0

Date 2021-06-16

Description Fitting (hierarchical) hidden Markov models
to daily share prices provided by <https://finance.yahoo.com/>.
See <https://github.com/loelschlaeger/fHMM#readme> for documentation and examples.

License GPL-3

Encoding UTF-8

Imports MASS, progress, Rcpp, tseries

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5.0)

RoxygenNote 7.1.1

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation yes

Author Lennart Oelschläger [aut, cre],
Timo Adam [aut],
Michels Rouven [aut]

Maintainer Lennart Oelschläger <lennart.oelschlaeger@uni-bielefeld.de>

Repository CRAN

Date/Publication 2021-06-16 16:00:02 UTC

R topics documented:

apply_viterbi ................................................................. 2
check_controls ............................................................... 3
check_decoding .............................................................. 3
check_estimation ............................................................ 4
check_saving ................................................................. 4
apply_viterbi

Description


Usage

apply_viterbi(data, fit, controls)

Arguments

data A list of processed data information.
fit A list of fitted model information.
controls A list of controls.

Value

A vector (in case of a HMM) or a matrix (in case of a hierarchical HMM) of decoded states.
**check_controls**  

---

**Description**  
This function checks the specification of controls.

**Usage**  
```
check_controls(controls)
```

**Arguments**  
- `controls` A list of controls.

**Details**  
See the vignettes on how to specify controls.

**Value**  
Checked version of controls.

---

**check_decoding**  

---

**Description**  
Summarizes and saves decoded states.

**Usage**  
```
check_decoding(decoding, data, controls)
```

**Arguments**  
- `decoding` A vector (in case of a hmm) or a matrix (in case of a hierarchical HMM) of decoded states.
- `data` A list of processed data information.
- `controls` A list of controls.

**Value**  
No return value. Creates output file "states.txt".
### check_estimation

**Estimation check**

**Description**

Summarizes and saves estimates.

**Usage**

```r
check_estimation(mods, lls, data, hessian, controls)
```

**Arguments**

- `mods` A list of fitted models in the different estimation runs.
- `lls` A vector of log-likelihood values of accepted `mods`.
- `data` A list of processed data information.
- `hessian` Hessian matrix of the estimated model.
- `controls` A list of controls.

**Value**

A list of fitted model information.

### check_saving

**Saving check**

**Description**

This function saves model results while checking for overwriting.

**Usage**

```r
check_saving(object = NULL, name = NULL, filetype, controls)
```

**Arguments**

- `object` An object to be saved.
- `name` A character, the name of the object to be saved.
- `filetype` A character, the filetype of the object to be saved.
- `controls` A list of controls.

**Value**

A boolean, determining whether saving is possible or not. If `filetype="rds"`, `object` is saved.
compute_ci

Confidence intervals

Description
Computes confidence intervals for the estimates.

Usage
compute_ci(fit, controls)

Arguments
- fit: A list of fitted model information.
- controls: A list of controls.

Value
A list containing the following elements:
- lb_ci_level: lower bound of the intervals
- estimate: estimates
- ub_ci_level: upper bound of the intervals
where ci_level is set in controls.

calculate_fs

Fine-scale chunk lengths

Description
Computes (flexible) fine-scale chunk lengths.

Usage
compute_fs(fs_time_horizon, T = NA, fs_dates = NA)

Arguments
- fs_time_horizon: Either a numeric or one of "w", "m", "q", "y", setting the fine-scale dimension.
- T: A numeric, the dimension of the coarse-scale process, default NA.
- fs_dates: A vector of dates of empirical fine-scale observations, default NA.

Value
A vector of fine-scale chunk sizes.
create_visuals | Visualization
---|---

**Description**
Calls functions for visualization of model results.

**Usage**
```r
create_visuals(data, fit, decoding, controls, events)
```

**Arguments**
- `data` A list of processed data information.
- `fit` A list of fitted model information.
- `decoding` A vector (in case of a HMM) or a matrix (in case of a hierarchical HMM) of decoded states.
- `controls` A list of controls.
- `events` A list of (historical, financial) events.

**Value**
No return value. Calls visualization functions `plot_sdd`, `plot_ts` and `pseudo_residuals`.

download_data | Data download
---|---

**Description**

**Usage**
```r
download_data(
  name = NA,
  symbol = NA,
  from = "1902-01-01",
  to = Sys.Date(),
  show_symbols = FALSE,
  path
)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A character, personal identifier for a stock, default NA.</td>
</tr>
<tr>
<td>symbol</td>
<td>A character, the stock's symbol, default NA.</td>
</tr>
<tr>
<td>from</td>
<td>A date, setting the lower data bound, default is &quot;1902-01-01&quot;.</td>
</tr>
<tr>
<td>to</td>
<td>A date, setting the upper data bound, default is the current date Sys.date().</td>
</tr>
<tr>
<td>show_symbols</td>
<td>A boolean, determining whether all saved symbols should be printed, default FALSE.</td>
</tr>
<tr>
<td>path</td>
<td>A character, setting the data saving path.</td>
</tr>
</tbody>
</table>

Details

symbol has to match the official symbol on [https://finance.yahoo.com](https://finance.yahoo.com). Once used stock symbols are saved in "stock_symbols.rds" in the folder "path/data". Values for from earlier than its default value are set to the default value.

Value

No return value. Downloaded data is saved as "name.csv" in the folder "path/data".

Examples

```r
### download 21st century DAX data
download_data(name="dax",symbol="^GDAXI",from=as.Date("2000-01-03"),path=tempdir())
```

exception  

Debugging

Description

Provides suggestions for debugging for a given exception code.

Usage

```r
exception(code)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>A character, the exception code.</td>
</tr>
</tbody>
</table>

Value

A list containing the following elements:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>exception code</td>
</tr>
<tr>
<td>response</td>
<td>message</td>
</tr>
<tr>
<td>debugging</td>
<td>suggestions for debugging</td>
</tr>
</tbody>
</table>
Example of a hidden Markov model (HMM) for financial data analysis.

## fit_hmm

Fit (hierarchical) hidden Markov models to financial data.

### Description

Performs data processing, fitting, state decoding and visualization.

### Usage

```r
fit_hmm(controls, events, sim_par)
```

### Arguments

- `controls`: A list of controls (optional).
- `events`: A list of (historical, financial) events (optional).
- `sim_par`: A list of model parameters for simulation in thetaList format, default NULL (optional).

### Details

Specify a model by setting parameters of the named list `controls` and passing it to `fit_hmm`. See the vignettes on how to specify `controls`.

### Value

No return value. Estimation results are saved in `"controls[["path"]]/models/controls[["id"]].`

### Examples

#### fitting a 2-state HMM with state-dependent t-distributions to simulated data

```r
controls = list(
    path = tempdir(),
    id = "test",
    model = "hmm",
    states = 2,
    sdds = "t",
    horizon = 200,
    fit = list("runs" = 10, "seed" = 1)
)
fit_hmm(controls)
```
**init_est**

**Initialisation**

**Description**
Samples initial parameter values for the estimation routine.

**Usage**
init_est(controls)

**Arguments**
- controls: A list of controls.

**Value**
A vector of parameters values in format thetaUncon.

**max_likelihood**

**Optimization**

**Description**
Maximizes the model’s log-likelihood function.

**Usage**
max_likelihood(data, controls)

**Arguments**
- data: A list of processed data information.
- controls: A list of controls.

**Details**
Uses nlm for numerical optimization.

**Value**
A list of fitted model information.
parameter_names

Description

Creates model parameter names.

Usage

parameter_names(controls, all)

Arguments

controls A list of controls.
all A boolean, determining whether all (all=TRUE) or only estimated (all=FALSE) names should be produced.

Value

Vector of model parameter names.

plot_ll

Description

Visualization of log-likelihood values

Plots log-likelihood values of the different estimation runs.

Usage

plot_ll(lls, controls)

Arguments

lls A vector of log-likelihood values.
controls A list of controls.

Value

No return value. Creates file "log_likelihoods.pdf" in "controls[["path"]]/models/controls[["id"]]".
**plot_sdd**  
*Visualization of estimated state-dependent distributions*

**Description**
Plots the estimated state-dependent distributions.

**Usage**
```
plot_sdd(controls, data, fit, decoding, colors)
```

**Arguments**
- **controls**: A list of controls.
- **data**: A list of processed data information.
- **fit**: A list of fitted model information.
- **decoding**: A vector (in case of a HMM) or a matrix (in case of a hierarchical HMM) of decoded states.
- **colors**: A matrix of colors for different states.

**Value**
No return value. Creates file "state_dependent_distributions.pdf" in "controls["path"]/models/controls["id"]".

---

**plot_ts**  
*Visualize decoded time-series*

**Description**
Visualize decoded time-series

**Usage**
```
plot_ts(controls, data, decoding, colors, events)
```

**Arguments**
- **controls**: A list of controls.
- **data**: A list of processed data information.
- **decoding**: A matrix of decoded states.
- **colors**: A matrix of colors for different states.
- **events**: A list of events.

**Value**
No return value, creates graphic in controls["path"]/models/controls["id"]
**process_data**  
*Data processing*

**Description**  
Calls functions for processing or simulating data.

**Usage**  
```
process_data(controls, sim_par)
```

**Arguments**
- **controls**: A list of controls.
- **sim_par**: A vector of model parameters for simulation.

**Value**  
A list of processed data information and on-screen information.

---

**pseudo_residuals**  
*Pseudo-residuals*

**Description**  
Computes and visualizes pseudo-residuals.

**Usage**  
```
pseudo_residuals(controls, data, fit, decoding)
```

**Arguments**
- **controls**: A list of controls.
- **data**: A list of processed data information.
- **fit**: A list of fitted model information.
- **decoding**: A vector (in case of a HMM) or a matrix (in case of a hierarchical HMM) of decoded states.

**Value**  
No return value. Creates files "pseudo_residuals.pdf" and "pseudos.rds" in "controls["path"]/models/controls["id"]".
**read_data**

*Read .csv-file*

**Description**

Reads financial data from .csv-file.

**Usage**

read_data(controls)

**Arguments**

- **controls**: A list of controls.

**Value**

A list containing the following elements:

- **data**: A matrix of data that is modeled.
- **data_raw**: A matrix of raw data.
- **data_fs_raw**: A matrix of raw fine-scale data.
- **data_cs_raw**: A matrix of raw coarse-scale data.
- **dates**: A vector of dates.
- **T_star**: A vector of fine-scale chunk sizes.

**simulate_data**

*Data simulation*

**Description**

Simulates data from a (hierarchical) hidden Markov model.

**Usage**

simulate_data(controls, sim_par)

**Arguments**

- **controls**: A list of controls.
- **sim_par**: A list of model parameters for simulation in thetaList format.
**simulate_data**

**Value**

A list containing the following elements:

- **data**: A matrix of simulated data.
- **states0**: A matrix of simulated hidden states.
- **thetaUncon0**: True parameters in format thetaUncon.
- **thetaCon0**: True parameters in format thetaCon.
- **thetaList0**: True parameters in format thetaList.
- **T_star**: A vector of fine-scale chunk sizes.
Index

apply_viterbi, 2
check_controls, 3
check_decoding, 3
check_estimation, 4
check_saving, 4
compute_ci, 5
compute_fs, 5
create_visuals, 6
download_data, 6
exception, 7
fit_hmm, 8
init_est, 9
max_likelihoood, 9
parameter_names, 10
plot_ll, 10
plot_sdd, 11
plot_ts, 11
process_data, 12
pseudo_residuals, 12
read_data, 13
simulate_data, 13