Package ‘DBHC’

Type Package

Title Sequence Clustering with Discrete-Output HMMs

Version 0.0.3

Date 2022-12-21

Author Gabriel Budel [aut, cre], Flavius Frasincar [aut]

Maintainer Gabriel Budel <gabysp_budel@hotmail.com>

Description Provides an implementation of a mixture of hidden Markov models (HMMs) for discrete sequence data in the Discrete Bayesian HMM Clustering (DBHC) algorithm. The DBHC algorithm is an HMM Clustering algorithm that finds a mixture of discrete-output HMMs while using heuristics based on Bayesian Information Criterion (BIC) to search for the optimal number of HMM states and the optimal number of clusters.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/gabybudel/DBHC

BugReports https://github.com/gabybudel/DBHC/issues

Imports seqHMM (>= 1.0.8), TraMineR (>= 2.0-7), reshape2 (>= 1.2.1), ggplot2 (>= 2.2.1), methods (>= 4.2.2)

NeedsCompilation no

Repository CRAN

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Date/Publication 2022-12-22 13:10:15 UTC

R topics documented:

assign.clusters ......................................................... 2
cluster.bic .............................................................. 3
Description

Assign sequences to cluster models that give the highest sequence-to-hmm likelihood. Used in `hmm.clust`.

Usage

assign.clusters(partition, memberships, sequences, smoothing = 1e-04)

Arguments

- **partition**: A list object with the partition, a mixture of HMMs. Each element in the list is an `hmm` object (see `build_hmm`).
- **memberships**: A matrix with cluster memberships for each sequence.
- **sequences**: An `stslist` object (see `seqdef`) of sequences with discrete observations.
- **smoothing**: Smoothing parameter for absolute discounting in `smooth.probabilities`.

Value

The updated matrix with cluster memberships for each sequence.

See Also

Used in main function for the DBHC algorithm `hmm.clust`. 
**cluster.bic**  

**HMM BIC**

**Description**

Compute the BIC of a single HMM given a threshold epsilon for counting parameters. Auxiliary function used in `size.search`.

**Usage**

```r
cluster.bic(hmm, eps = 0.001)
```

**Arguments**

- `hmm` An `hmm` object (see `build.hmm`).
- `eps` A threshold epsilon for counting parameters.

**Value**

The BIC of `hmm`.

**See Also**

Used in `size.search`.

---

**count.parameters**  

**Count HMM Parameters**

**Description**

Count the number of parameters in an HMM larger than a small number epsilon. Auxiliary function used in `partition.bic` and `cluster.bic`.

**Usage**

```r
count.parameters(hmm, eps = 0.001)
```

**Arguments**

- `hmm` An `hmm` object (see `build.hmm`).
- `eps` A threshold epsilon for counting parameters.

**Value**

The number of parameters larger than `eps`. 
See Also

Used in `partition.bic` and `cluster.bic`.

emission.heatmap  

Description

Plots a heatmap of an HMM’s emission probabilities.

Usage

```r
emission.heatmap(emission, base_size = 10)
```

Arguments

- `emission` A matrix with emission probabilities (see also `build_hmm`).
- `base_size` Numerical, a size parameter for the plots made using `ggplot2` (see `theme`), default = 10.

See Also

See `hmm.clust` for an example.

hmm.clust  

Description

Implementation of the DBHC algorithm, an HMM clustering algorithm that finds a mixture of discrete-output HMMs. The algorithm uses heuristics based on BIC to search for the optimal number of hidden states in each HMM and the optimal number of clusters.

Usage

```r
hmm.clust(
  sequences,
  id = NULL,
  smoothing = 1e-04,
  eps = 0.001,
  init.size = 2,
  alphabet = NULL,
  K.max = NULL,
  log_space = FALSE,
  print = FALSE,
  seed.size = 3
)
```
Arguments

sequences  An stslist object (see seqdef) of sequences with discrete observations or a data.frame.
id  A vector with ids that identify the sequences in sequences.
smoothing  Smoothing parameter for absolute discounting in smooth.probabilities.
eps  A threshold epsilon for counting parameters in count.parameters.
init.size  The number of HMM states in an initial HMM.
alphabet  The alphabet of output labels, if not provided alphabet is taken from stslist object (see seqdef).
K.max  Maximum number of clusters, if not provided algorithm searches for the optimal number itself.
log_space  Logical, parameter provided to fit_model for whether to use optimization in log space or not.
print  Logical, whether to print intermediate steps or not.
seed.size  Seed size, the number of sequences to be selected for a seed

Value

A list with components:

sequences  An stslist object of sequences with discrete observations.
id  A vector with ids that identify the sequences in sequences.
cluster  A vector with found cluster memberships for the sequences.
partition  A list object with the partition, a mixture of HMMs. Each element in the list is an hmm object.
memberships  A matrix with cluster memberships for each sequence.
n.clusters  Numerical, the found number of clusters.
sizes  A vector with the number of HMM states for each cluster model.
bic  A vector with the BICs for each cluster model.

Examples

## Simulated data
library(seqHMM)
output.labels <- c("H", "T")

# HMM 1
states.1 <- c("A", "B", "C")
transitions.1 <- matrix(c(0.8,0.1,0.1,0.1,0.8,0.1,0.1,0.1,0.8), nrow = 3)
rownames(transitions.1) <- states.1
colnames(transitions.1) <- states.1
densities.1 <- matrix(c(0.5,0.75,0.25,0.5,0.25,0.75), nrow = 3)
rownames(densities.1) <- states.1
colnames(densities.1) <- output.labels
initials.1 <- c(1/3,1/3,1/3)
# HMM 2
states.2 <- c("A", "B")
transitions.2 <- matrix(c(0.75, 0.25, 0.25, 0.75), nrow = 2)
rownames(transitions.2) <- states.2
colnames(transitions.2) <- states.2
emissions.2 <- matrix(c(0.8, 0.6, 0.2, 0.4), nrow = 2)
rownames(emissions.2) <- states.2
colnames(emissions.2) <- output.labels
initials.2 <- c(0.5, 0.5)

# Simulate
hmm.sim.1 <- simulate_hmm(n_sequences = 100,
  initial_probs = initials.1,
  transition_probs = transitions.1,
  emission_probs = emissions.1,
  sequence_length = 25)
hmm.sim.2 <- simulate_hmm(n_sequences = 100,
  initial_probs = initials.2,
  transition_probs = transitions.2,
  emission_probs = emissions.2,
  sequence_length = 25)
sequences <- rbind(hmm.sim.1$observations, hmm.sim.2$observations)
n <- nrow(sequences)

# Clustering algorithm
id <- paste0("K-", 1:n)
rownames(sequences) <- id
sequences <- sequences[sample(1:n, n),]
res <- hmm.clust(sequences, id = rownames(sequences))

############################################################

## Swiss Household Data
data("biofam", package = "TraMineR")

# Clustering algorithm
sequences <- seqdef(biofam[,10:25], alphabet = 0:7, states = new.alphabet)

## Not run:
res <- hmm.clust(sequences)

# Heatmaps
cluster <- 1 # display heatmaps for cluster 1
transition.heatmap(res$partition[[cluster]]$transition_probs,
  res$partition[[cluster]]$initial_probs)
emission.heatmap(res$partition[[cluster]]$emission_probs)

## End(Not run)
## A smaller example, which takes less time to run

```r
subset <- sequences[sample(1:nrow(sequences), 20, replace = FALSE),]

# Clustering algorithm, limiting number of clusters to 2
res <- hmm.clust(subset, K.max = 2)

# Number of clusters
print(res$n.clusters)

# Table of cluster memberships
table(res$memberships[, "cluster")

# BIC for each number of clusters
print(res$bic)

# Heatmaps
cluster <- 1  # display heatmaps for cluster 1
transition.heatmap(res$partition[[cluster]]$transition_probs,
                  res$partition[[cluster]]$initial_probs)
emission.heatmap(res$partition[[cluster]]$emission_probs)
```

---

**model.ll**

### Get HMM Log Likelihood

**Description**

Get the log likelihood of an HMM object and check if it is feasible (i.e., contains no illegal emissions). Auxiliary function used in `partition.bic`.

**Usage**

```r
model.ll(hmm)
```

**Arguments**

- `hmm` An `hmm` object (see `build_hmm`).

**Value**

The log likelihood of the `hmm` object, print warning if model is infeasible (i.e., if the log likelihood is evaluated for a sequence that contains emissions that are assigned probability 0 in the `hmm` object).

**See Also**

Used in `partition.bic`. 
### partition.bic

**Partition BIC**

**Description**

Compute the BIC of a partition given a threshold epsilon for counting parameters. Auxiliary function used in `hmm.clust`.

**Usage**

```r
partition.bic(partition, eps = 0.001)
```

**Arguments**

- `partition`: A list object with the partition of HMMs, a mixture of HMMs.
- `eps`: A threshold epsilon for counting parameters in `count.parameters`.

**Value**

The BIC of the partition.

**See Also**

Used in main function for the DBHC algorithm `hmm.clust`.

### select.seeds

**Seed Selection Procedure**

**Description**

Seed selection procedure of the DBHC algorithm, also invokes size search algorithm for seed in `size.search`. Used in `hmm.clust`.

**Usage**

```r
select.seeds(
  sequences,
  log_space = FALSE,
  K,
  seed.size = 3,
  init.size = 2,
  print = FALSE,
  smoothing = 1e-04
)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequences</td>
<td>An <code>stslist</code> object (see <code>seqdef</code>) of sequences with discrete observations.</td>
</tr>
<tr>
<td>log_space</td>
<td>Logical, parameter provided to <code>fit_model</code> for whether to use optimization in log space or not.</td>
</tr>
<tr>
<td>K</td>
<td>The number of seeds to select, equal to the number of clusters in a partition.</td>
</tr>
<tr>
<td>seed.size</td>
<td>Seed size, the number of sequences to be selected for a seed.</td>
</tr>
<tr>
<td>init.size</td>
<td>The number of HMM states in an initial HMM.</td>
</tr>
<tr>
<td>print</td>
<td>Logical, whether to print intermediate steps or not.</td>
</tr>
<tr>
<td>smoothing</td>
<td>Smoothing parameter for absolute discounting in <code>smooth.probabilities</code>.</td>
</tr>
</tbody>
</table>

Value

A partition as a list object with HMMs for the selected seeds.

See Also

Used in main function for the DBHC algorithm `hmm.clust`.

Description

Compute the sequence-to-HMM likelihood of an HMM evaluated for a single sequence and check if the sequence contains emissions that are not possible according to the HMM. Auxiliary function used in `select.seeds` and `assign.clusters`.

Usage

```r
seq2hmm.ll(hmm)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hmm</td>
<td>An hmm object (see <code>build_hmm</code>) containing a single sequence.</td>
</tr>
</tbody>
</table>

Value

The log likelihood of the sequence contained in `hmm`, value will be set to minus infinity if the sequence contains illegal emissions.

See Also

Used in `select.seeds` and `assign.clusters`. 
size.search  

Size Search Algorithm

Description

The size search algorithm finds the optimal number of HMM states for a set of sequences and returns both the optimal `hmm` object and the corresponding number of hidden states. Used in `select.seeds`.

Usage

```r
size.search(sequences, log_space = FALSE, print = FALSE)
```

Arguments

- `sequences`: An `stslist` object (see `seqdef`) of sequences with discrete observations.
- `log_space`: Logical, parameter provided to `fit_model` for whether to use optimization in log space or not.
- `print`: Logical, whether to print intermediate steps or not.

Value

A list with the optimal number of HMM states and the optimal `hmm` object.

See Also

Used in the DBHC seed selection procedure in `select.seeds`.

smooth.hmm

Smooth HMM Parameters

Description

Smooth the parameters of an HMM using absolute discounting given a threshold epsilon. Auxiliary function used in `select.seeds`, `assign.clusters`, and `hmm.clust`.

Usage

```r
smooth.hmm(hmm, smoothing = 1e-04)
```

Arguments

- `hmm`: A raw `hmm` object (see `build_hmm`).
- `smoothing`: Smoothing parameter for absolute discounting in `smooth.probabilities`.
smooth.probabilities

Value
An hmm object with smoothed probabilities.

See Also
Used in select.seeds, assign.clusters, and main function for the DBHC algorithm hmm.clust.

smooth.probabilities Smooth Probabilities

Description
Smooth a vector of probabilities using absolute discounting. Auxiliary function used in smooth.hmm.

Usage
smooth.probabilities(probs, smoothing = 1e-04)

Arguments
probs A vector of raw probabilities.
smoothing Smoothing parameter for absolute discounting.

Value
A vector of smoothed probabilities.

See Also
Used in smooth.hmm.

transition.heatmap Heatmap Transition Probabilities

Description
Plots a heatmap of an HMM's initial and transition probabilities.

Usage
transition.heatmap(transition, initial = NULL, base_size = 10)
transition.heatmap

Arguments

- **transition**: A matrix with transition probabilities (see also `build_hmm`).
- **initial**: An (optional) vector of initial probabilities.
- **base_size**: Numerical, a size parameter for the plots made using `ggplot2` (see `theme`), default = 10.

See Also

See `hmm.clust` for an example.
Index

assign.clusters, 2, 9–11
build_hmm, 2–4, 7, 9, 10, 12
cluster.bic, 3, 3, 4
count.parameters, 3, 5, 8
emission.heatmap, 4
fit_model, 5, 9, 10
hmm.clust, 2, 4, 4, 8–12
model.ll, 7
partition.bic, 3, 4, 7, 8
select.seeds, 8, 9–11
seq2hmm.ll, 9
seqdef, 2, 5, 9, 10
size.search, 3, 8, 10
smooth.hmm, 10, 11
smooth.probabilities, 2, 5, 9, 10, 11
theme, 4, 12
transition.heatmap, 11