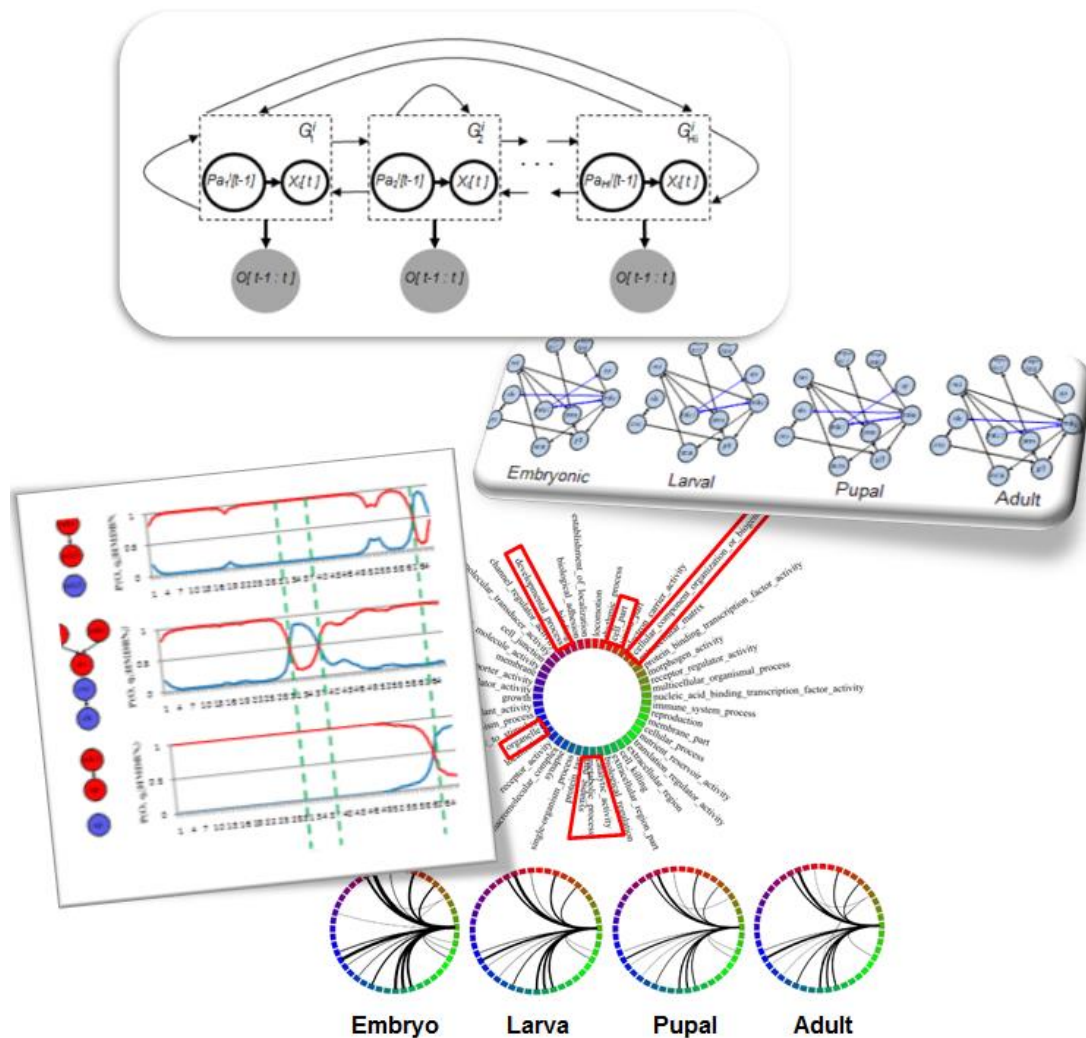

HIDDEN MARKOV INDUCED DYNAMIC BAYESIAN NETWORK (HMDBN)

SHIJIA ZHU, YADONG WANG



Contents

Installation Requirement	1
A Toy Example	2
hmdbn_structEM	4
hmdbn_demonstrate.....	5
hmdbn_hiddenGraphs_and_BWBIC_node	6
hmdbn_updateA	8
hmdbn_mhmm_A	10
hmdbn_fwdback	11
hmdbn_prob_node.....	13
hmdbn_BWBIC_score	14
hmdbn_log_prob_node	16
hmdbn_learn_params	17
hmdbn_compute_counts	18

Installation Requirement

Details

HMDBN was implemented in matlab based on the Bayes Net Toolbox package, written by Kevin Murphy, 1997-2002, <https://code.google.com/p/bnt/>

References

‘The Bayes Net Toolbox for Matlab Computing’, Kevin P. Murphy, *Science and Statistics*, 2001

See Also

A Toy Example

Description

A simple example for learning a time-evolving DBN from time series dataset.

Examples

```
data=load('data.txt');
time1 = cputime;
[dag EmissionGraph_Ps SampleDistribution] = hmdbn_structEM(data);
ElapsedTime = cputime - time1;
hmdbn_demonstrate(data,EmissionGraph_Ps,SampleDistribution);
```

Details

The main function ‘hmdbn_structEM’ implemented a Structural Expectation Maximization (SEM) algorithm to learn the time-evolving DBN; the simulated dataset in this example includes 500 observations, which come from a time-evolving DBN with three nodes and two single-edge changes across three time segments. The true time-varying network topology is shown in Fig. 1a. The function ‘hmdbn_demonstrate’ shows the prediction results in Fig. 1b.

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_structEM, hmdbn_demonstrate

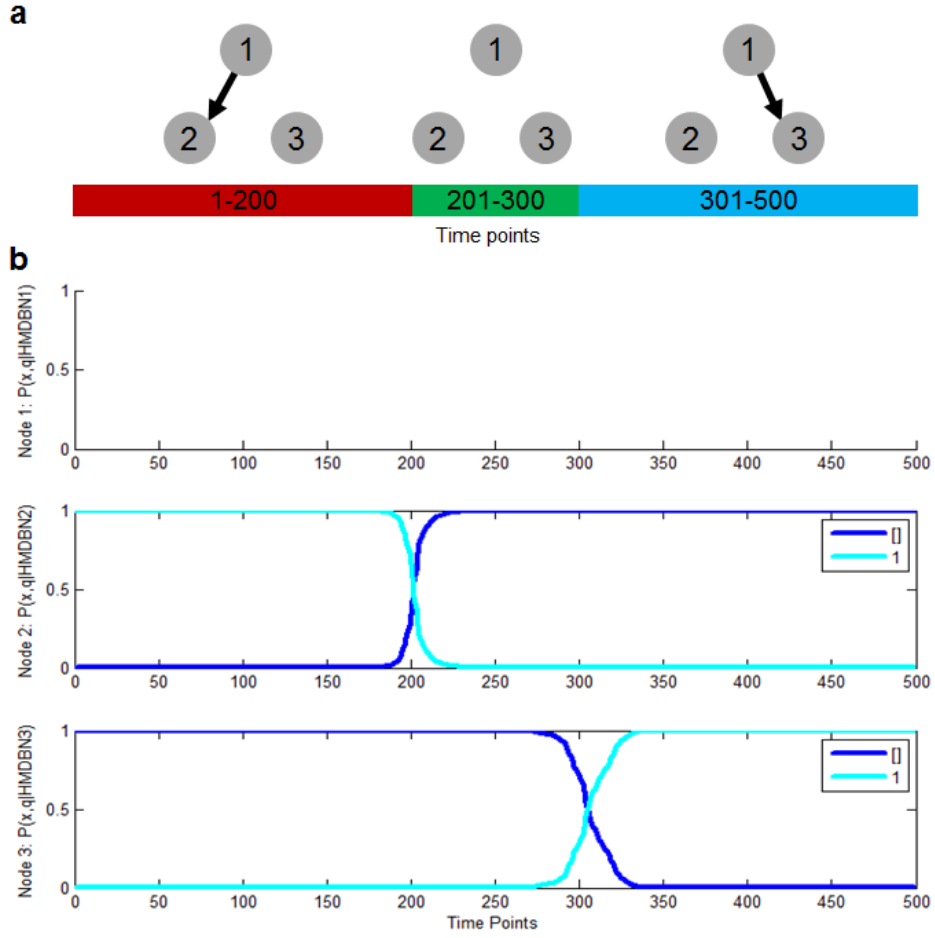


Fig. 1. A toy example. (a) The true time-evolving network topology for the simulated dataset. The simulated dataset includes 500 observations. Three different networks happen in time segment 1-200, 201-300 and 301-500, respectively. (b) The HMDBN-predicted probability of each network topology along time points. Horizontal axis denotes time points. Vertical axis represents the probability density. The legend on the right top corner gives the recovered parents for each node. The curves with different colors respectively denote the probability density over networks in the same colors on the right top corner.

hmdbn_structEM

Description

The main function for learning a time-evolving DBN.

Usage

```
[seeddag    hiddenGraph_Ps    SampleDistribution]    =    hmdbn_structEM  
(TimeSeriesData)
```

Arguments

TimeSeriesData the input time series dataset; TimeSeriesData(i,m) represents the value of node i in observation m.

Details

The Structural Expectation Maximization algorithm was used to learn a time-evolving DBN.

Value

seeddag	the resulting global stationary DBN structure of all time segments; seeddag(i,j) = 1 denotes the edge (i→j);
hiddenGraph_Ps	the parent node set in hidden graphs of HMDBN for each node;
SampleDistribution	the weighted sample for each hidden graph along time points.

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_demonstrate

Description

The function for demonstrating a time-evolving DBN.

Usage

hmdbn_demonstrate (data, hiddenGraph_Ps, SampleDistribution)

Arguments

data	the input time series dataset;
hiddenGraph_Ps	the parent node set in hidden graphs of HMDBN for each node;
SampleDistribution	the weighted sample for each hidden graph along time points.

Details

The function demonstrates the prediction results by function ‘hmdbn_structEM’.

Value

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_structEM

hmdbn_hiddenGraphs_and_BWBIC_node

Description

The function for calculating the BWBIC score of HMDBN for node j .

Usage

```
[score SampleDistribution hiddenGraph_g2_Ps] =  
hmdbn_hiddenGraphs_and_BWBIC_node(j, ps, ns, data, SeedSampleDistribution)
```

Arguments

j	the j -th node;
ps	the global parents of node j in all hidden graphs;
ns	the node size for each node, i.e. the number of discrete states of the variable for each node;
$data$	the observation dataset; $data(i,m)$ represents the value of node i in observation m ;
$SeedSampleDistribution$	weighted samples for HMDBN of node j ; the HMDBN has only two nodes i and j , and two hidden graphs, one without edges and the other with only one edge $i \rightarrow j$.

Details

First, according to $SeedSampleDistribution$, this function identifies the putative hidden graphs and set the initial weighted sample distribution for each putative hidden graph;

Second, the function 'hmdbn_updateA' iteratively updates the transition matrix and updates the weighted samples;

Third, the function 'hmdbn_BWBIC_score' calculates the BWBIC score.

Value

$score$	the obtained BWBIC score of HMDBN for node j ;
$hiddenGraph_g2_Ps$	parent nodes in each hidden graph of HMDBN for node j .
$SampleDistribution$	the learned weighted sample distribution for each hidden graph.

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_updateA; hmdbn_BWBIC_score

hmdbn_updateA

Description

The function for iteratively updating the transition matrix.

In the following description, we assume $O(t)$ = observation, $Q(t)$ = hidden state.

Usage

`[alpha, beta, gamma, xi, transmat] = hmdbn_updateA (gamma, xi, transmat, obslik, init_state_distrib)`

Arguments

gamma	$\gamma(i,t) = \Pr(Q(t) = i \mid O(1:T))$, representing the weighted sample;
xi	$\xi(i,j,t-1) = \Pr(Q(t-1) = i, Q(t) = j \mid O(1:T))$, which is used to update the transition matrix;
transmat	The transition matrix, where $\text{transmat}(i,j) = \Pr(Q(t) = j \mid Q(t-1) = i)$, representing the transition probability from hidden graph G_i to G_j ;
obslik	The emission probability, where $\text{obslik}(i,t) = \Pr(O(t) \mid Q(t) = i)$, representing the probability of observation $O(t)$ conditioning on hidden graph G_i ;
init_state_distrib	transition probabilities for initial hidden graphs, $\text{init_state_distrib}(i) = \Pr(Q(1) = i)$.

Details

The function iteratively updates the transition matrix. First, according to the current values of gamma and xi, the function ‘hmdbn_mhmm_A’ re-estimates the transition matrix; next, according to the updated transition matrix, the function ‘hmdbn_fwdback’ re-calculates the values of gamma and xi. These two steps repeat iteratively, until the transition matrix converges.

Value

alpha	$\alpha(i,t) = \Pr(O(1:t), Q(t) = i)$;
beta	$\beta(i,t) = \Pr(O(t+1:T) \mid Q(t)=i)$;
gamma	$\gamma(i,t) = \Pr(Q(t) = i \mid O(1:T))$, which is the updated weighted sample;
xi	$\xi(i,j,t-1) = \Pr(Q(t-1) = i, Q(t) = j \mid O(1:T))$, which is the

transmat updated xi;
 the updated transition matrix.

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_mhmm_A, hmdbn_fwdback, hmdbn_prob_node

hmdbn_mhmm_A

Description

The function for updating the transition matrix.

Usage

```
transmatNew = my_mhmm_A(gamma, xi)
```

Arguments

gamma	$\text{gamma}(i,t) = \Pr(Q(t) = i \mid O(1:T))$, representing the weighted sample;
xi	$\text{xi}(i,j,t-1) = \Pr(Q(t-1) = i, Q(t) = j \mid y(1:T))$, which is used to update the transition matrix;

Details

According to the re-estimation formulas, estimate the transition matrix.

Value

transmatNew the re-estimated transition matrix.

Author(s)

Shijia Zhu, <shja.zhu@gmail.com>; Yadong Wang <ydwang@hit.edu.cn>

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_fwdback

Description

The function for calculating weighted samples.

In the following description, we assume $O(t)$ = observation, $Q(t)$ = hidden state.

Usage

`[alpha, beta, gamma, loglik, xi] = hmdbn_fwdback (init_state_distrib, transmat, obslik)`

Arguments

<code>init_state_distrib</code>	transition probabilities for initial hidden graphs, $\text{init_state_distrib}(i) = \Pr(Q(1) = i)$;
<code>transmat</code>	The transition matrix, where $\text{transmat}(i,j) = \Pr(Q(t) = j \mid Q(t-1) = i)$, representing the transition probability from hidden graph G_i to G_j ;
<code>obslik</code>	The emission probability, where $\text{obslik}(i,t) = \Pr(O(t) \mid Q(t) = i)$, representing the probability of observation $O(t)$ conditioning on hidden graph G_i .

Details

This function implements the forward-backward algorithm.

Value

<code>alpha</code>	$\alpha(i,t) = \Pr(O(1:t), Q(t) = i)$;
<code>beta</code>	$\beta(i,t) = \Pr(O(t+1:T) \mid Q(t)=i)$;
<code>gamma</code>	$\gamma(i,t) = \Pr(Q(t)=i \mid O(1:T))$, which is the learned weighted sample;
<code>loglik</code>	$\text{loglik} = \log \Pr(O(1:T))$;
<code>xi</code>	$\xi(i,j,t-1) = \Pr(Q(t-1)=i, Q(t)=j \mid y(1:T))$.

Author(s)

Shijia Zhu, <shja.zhu@gmail.com>; Yadong Wang <ydwang@hit.edu.cn>

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_prob_node

hmdbn_prob_node

Description

The function for computing the emission probability of one hidden graph.
Suppose the hidden graph is G_j , where j and ps are the target node and its parent node set, respectively; suppose $O(t)$ = observation, $Q(t)$ = hidden state.

Usage

obslik = hmdbn_prob_node (CPD, self_ev, pev)

Arguments

CPD	the parameters for the network G_j , which is a Conditional Probability Distribution (CPD).
self_ev	the observation data for node j ;
pev	the observation data for the parent node set ps .

Details**Value**

obslik	the emission probability, where $\text{obslik}(j,t) = \Pr(O(t) \mid Q(t) = j)$, representing the probability of observation $O(t)$ conditioning on hidden graph G_j .
--------	--

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_BWBIC_score

Description

The function for calculating the BWBIC score of a candidate hidden graph in HMDBN for node j .

Usage

[score CPD] = hmdbn_BWBIC_score (j, ps, ns, data, gammaG)

Arguments

j	the j -th node;
ps	the parent node set of node j in the hidden graph ($ps \rightarrow j$);
ns	node sizes, i.e. the number of discrete states of the variable for each node;
data	the observation dataset; data(i, m) represents the value of node i in observation m ;
gammaG	the input weighted samples.

Values

score	the resulting BWBIC score for hidden graphs of HMDBN with node j and its parent node set ps ;
CPD	the parameters for the network G_j , which is a Conditional Probability Distribution (CPD).

Details

The BWBIC score is an approximation for the objective function of Structural EM algorithm. It is composed of two parts: the first part is the maximized value of the likelihood function; the second part is the penalty for model complexity, including the number of edges and the number of hidden graphs.

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_log_prob_node

hmdbn_log_prob_node

Description

This function for computing the value of $\gamma \log(\Pr(O|G_j, \text{CPD}))$.
Suppose the hidden graph is G_j , where j and ps are the target node and its parent node set, respectively.

Usage

`L = hmdbn_log_prob_node (CPD, self_ev, pev, gamma);`

Arguments

CPD	the Maximum Likelihood Estimate of parameters for the candidate hidden graph G_j ;
self_ev	the observation data for node j ;
pev	the observation data for the parent node set ps .
gamma	$\gamma(j,t) = \Pr(Q(t) = j, O(1:T))$, which is the weighted sample, representing the probability that the observation data at time t is generated by the hidden graph G_j .

Details

The function $\gamma \log(\Pr(O|G, \text{CPD}))$ is one part of BWBIC score, where CPD is the Maximum Likelihood Estimate of parameters for the candidate hidden graph G_j .

Value

`L` the output value of function $\gamma \log(\Pr(O|G_j, \text{CPD}))$.

Author(s)

Shijia Zhu, <shja.zhu@gmail.com>; Yadong Wang <ydwang@hit.edu.cn>

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_learn_params

Description

The function for estimating the network parameters of one hidden graph.
Suppose the hidden graph is G_j , where j and ps are the target node and its parent node set, respectively.

Usage

CPD = hmdbn_learn_params (fam, ns, data, gamma)

Arguments

fam	the combination of ps and j : $fam = [ps\ j]$;
ns	node sizes, i.e. the number of discrete states of the variable for each node;
data	the observation dataset; $data(i,m)$ represents the value of node i in observation m ;
gamma	$gamma(j,t) = \Pr(Q(t) = j, O(1:T))$, which is the weighted sample, representing the probability that the observation data at time t is generated by the hidden graph G_j .

Details

Value

CPD	the Maximum Likelihood Estimate of Conditional Probability Distribution, i.e., the estimated parameters for the network G_j .
-----	---

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also

hmdbn_compute_counts

hmdbn_compute_counts

Description

The function organizes the observation dataset for one hidden graph into a conditional probability table.

Suppose the hidden graph is G_j , where j and ps are the target node and its parent node set, respectively.

Usage

```
count = hmdbn_compute_counts(local_data, sz, gamma)
```

Arguments

local_data	the observation dataset for nodes, ps and j
sz	the node sizes for ps and j ;
gamma	$\gamma(j,t) = p(Q(t)=j, O(1:T))$, which is the weighted sample, representing the probability that the observation data at time t is generated by the hidden graph G_j .

Details

Value

count	the learned Conditional Probability Table.
-------	--

Author(s)

Shijia Zhu, < shja.zhu@gmail.com >; Yadong Wang < ydwang@hit.edu.cn >

References

‘Hidden Markov induced Dynamic Bayesian Network for recovering time evolving gene regulatory networks’, Shijia Zhu and Yadong Wang, 2014

See Also