Causal Learner User's Manual
Public domain version 1.0
Release date: 24/10/2020

School of Computer Science and Technology,

Anhui University,

Hefei, Anhui, 230601, China

Authors:

Zhaolong Ling (zlling@ahu.edu.cn)

Kui Yu (yukui@hfut.edu.cn)

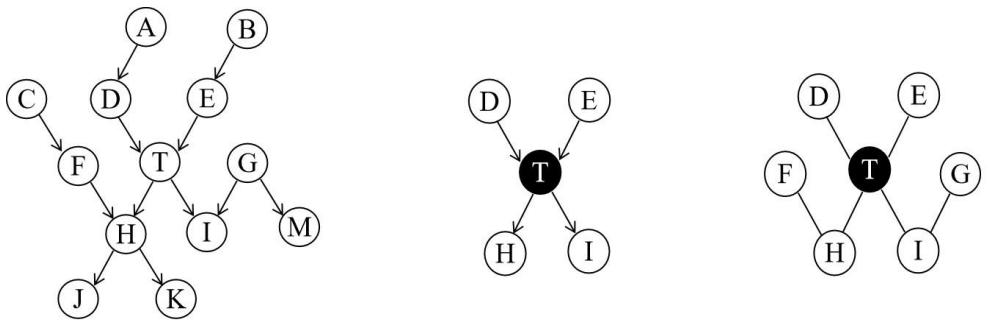
Causal Learner is an easy-to-use open-source toolbox for causal structure learning and Markov blanket learning, including simulated Bayesian network data generation functions, causal structure and Markov blanket learning algorithms, and algorithm evaluation functions.

Contents

1. Toolbox overview

1.1 Introduction

Causal Learner is a toolbox for learning causal structure and Markov blanket (MB) from data. It integrates functions for generating simulated Bayesian network (BN) data, a set of state-of-the-art global causal structure learning algorithms, a set of state-of-the-art local causal structure learning algorithms, a set of state-of-the-art MB learning algorithms, and functions for evaluating algorithms. Figure 1 gives examples of a global causal structure, local causal structure, and MB (T in black is a target node). The data generation part of Causal Learner is written in R, and the rest of Causal Learner is written in MATLAB. Causal Learner aims to provide researchers and practitioners with an open-source platform for causal learning from data and for the development and evaluation of new causal learning algorithms.



(a) Global causal structure (b) Local causal structure (c) Markov blanket

Figure 1: Examples of a global causal structure, local causal structure, and Markov blanket

1.2 Architecture

Figure 2 shows the hierarchical architecture of Causal Learner, in comparison with Causal Explorer[1]. As Causal Explorer represents the state-of-the-art ten years ago, it does not contain many new algorithms, and it does not have a data generation or evaluation component. By contrast, Causal Learner conceives a more ambitious blueprint. It aims to support the entire causal structure and MB learning procedure, including data generation, state-of-the-art algorithms, and algorithm evaluation.

[1] Alexander Statnikov, Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. Causal explorer: A matlab library of algorithms for causal discovery and variable selection for classification. Causation and Prediction Challenge Challenges in Machine Learning, Volume 2, pages 267–278, 2010.

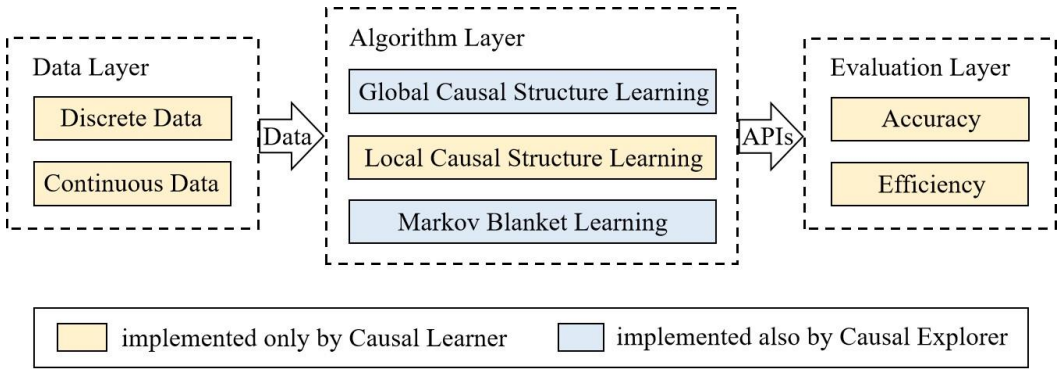


Figure 2: The architecture of Causal Learner

The Causal Learner website is at http://bigdata.ahu.edu.cn/causal-learner. The main directory of Causal Learner contains 7 folders and 4 MATLAB files, and the details of each document is shown in Table 1.

Type	Document	Description
Folder	alg_GCS	Global causal structure learning algorithms
	alg_LCS	Local causal structure learning algorithms
	alg_MB	Markov blanket learning algorithms
	common	Functions that all algorithms need to call in common
	data	Simulated data sets from various benchmark Bayesian networks, and true network structures
	evaluation	Algorithm evaluation functions
	Gen_Data	Simulated data sets generation functions
File	Causal_Learner.m	Interface of causal learner
	example_GCS.m	An example of global causal structure learning
	example_LCS.m	An example of local causal structure learning
	example_MB.m	An example of Markov blanket learning

Table 1: Details of the documents in Causal Learner

1.3 Configuration

To run Causal Learner, it is required that

(1) Windows 10 operating system;

(2) R 3.60[1] or above version, and R studio[2];

(3) MATLAB 2016a, or above to be installed.

[1] https://mirrors.tuna.tsinghua.edu.cn/CRAN/

[2] https://rstudio.com/products/rstudio/download/

2. Data

Causal Learner uses the documents in Gen_Data folder to generate simulated BN data sets (written in R), and the details of each benchmark BN[3] are shown in Table 2. The main directory of Gen_Data contains 1 folder and 2 R files, and the details of each document is shown in Table 3.

	Name	#Nodes	#Arcs	Name	#Nodes	#Arcs
Discrete Bayesian Network	CANCER	5	4	BARLEY	48	84
	EARTHQUAKE	5	4	HAILFINDER	56	66
	SURVEY	6	6	HEPAR II	70	123
	ASIA	8	8	WIN95PTS	76	112
	SACHS	11	17	PATHFINDER	109	195
	CHILD	20	25	ANDES	223	338
	INSURANCE	27	52	DIABETES	413	602
	WATER	32	66	PIGS	441	592
	MILDEW	35	46	LINK	724	1125
	ALARM	37	46	MUNIN (4 subnetworks)	186 -1041	273 -1388
Continuous Bayesian Network	SANGIOVESE	15	55	ECOLI70	46	70
	MEHRA	24	71	MAGIC-IRRI	64	102
	MAGIC-NIAB	44	66	ARTH150	107	150

Table 2: Benchamark Bayesian networks (#Nodes: the number of nodes, #Arcs: the number of arcs)

Type	Document	Description
Folder	RDS	Detailed information of Bayesian networks
File	Gen_Data.R	Interface of data generation
	example.R	An example of generating data

Table 3: Details of the documents in Gen_Data

To generate data, users only need to modify the parameters in example.R, and then run example.R to obtain a data set that simulates any BN with any data samples. For example, if users need to generate a simulated ALARM BN data set with 5000 data samples, please modify the parameters of data name (users can use one of the algorithm names in Table 2) and data samples in example.R, for example in Figure 3. Then run example.R, the generated data set will be stored in the data folder in the upper-level directory, alarm_5000.txt is the data set that simulates 5000 data samples of alarm BN. Meanwhile, the true BN structure will be also generated and stored in the data folder in the form of graph, such as alarm_graph.txt.

```
38 source('Gen_Data.R')
39
40 data_name<-"alarm"
41
42 data_samples<-5000
43
44 Gen_Data(data_name,data_samples)
```

Figure 3: An example of generating data set

[3] https://www.bnlearn.com/bnrepository

3. Algorithm

Causal Learner implements 7 global causal structure learning algorithms, 4 local causal structure learning algorithms, and 15 MB learning algorithms (written in MATLAB). Table 4 lists all of these algorithms.

To ensure the correctness of the algorithms implemented in Causal Learner, unless the original implementations of the algorithms are not released, we always try to integrate the original versions rather than re-implement them. Additionally, we used the same data to evaluate the algorithms in Causal Learner. Compared with Causal Explorer, the results of Causal Learner are comparable in accuracy and much more efficient.

Global Causal Structure Learning			Local Causal Structure Learning			Markov Blanket Learning		
Algorithms	Learner	Explorer	Algorithms	Learner	Explorer	Algorithms	Learner	Explorer
SCA	◦	•	PCD-by-PCD	•	◦	GS	•	•
PC	•	•	MB-by-MB	•	◦	IAMB	•	•
TPDA	◦	•	CMB	•	◦	interIAMB	•	•
GES	•	◦	LCS-FS	•	◦	IAMBnPC	•	•
GSBN	•	◦				interIAMBnPC	•	•
MMHC	•	•				Fast-IAMB	•	•
PC-stable	•	◦				FBED	•	◦
F2SL-c	•	◦				MMMB	•	•
F2SL-s	•	◦				HITON-MB	•	•
						PCMB	•	◦
						IPCMB	•	◦
						MBOR	•	◦
						STMB	•	◦
						BAMB	•	◦
						EEMB	•	◦

Table 4: Algorithms included in (indicated by •) and absent from (indicated by ◦) Causal Learner and Causal Explorer

3.1 Global causal structure learning algorithms

Causal Learner has implemented 7 global causal structure learning algorithms, the full name of each algorithm is shown in Table 5, and then the references of all algorithms.

- ★ GSBN - Grow/Shrink Bayesian network algorithm[2]
- ★ GES - Greedy Equivalence Search algorithm[3]
- ★ PC - PC algorithm[4]
- ★ MMHC - Max-Min Hill-Climbing algorithm[5]
- ★ PCstable - PC-stable algorithm[6]
- ★ F2SL_c - Feature Selection-based Structure Learning using independence tests (F2SL-c) algorithm[7]
- ★ F2SL_s - Feature Selection-based Structure Learning using score functions (F2SL-s) algorithm[7]

Table 5: Full name of all global causal structure learning algorithms

[2] Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. *In Advances in neural information processing systems*, pages 505–511, 2000.

[3] David Maxwell Chickering. 2003. Optimal Structure Identification With Greedy Search. *Journal of Machine Learning Research* 3 (2003), 507–554.

[4] Peter Spirtes, Clark N Glymour, and Richard Scheines. Causation, prediction, and search. *MIT press*, 2000.

[5] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[6] Diego Colombo and Marloes H. Maathuis. Order-independent constraint based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

[7] Kui Yu, Zhaolong Ling, Lin Liu, Hao Wang, and Jiuyong Li. Feature Selection for Efficient Local-to-Global Bayesian Network Structure Learning. *ACM Transactions on Intelligent Systems and Technology* (in press).

The input and output parameter types and quantities are the same for all global causal structure learning algorithms as shown in Table 6.

Inputs (four parameters):	1st input = algorithm name
	2nd input = data set
	3rd input = type of data set (discrete or continuous)
	4th input = significance level
Outputs (two parameters):	1st output = learned global causal structure
	2nd output = running time

Table 6: Inputs and outputs of global structure learning algorithms

Inputs (four parameters):

1st input = algorithm name

Users can use one of the algorithm names in Table 5 as the 1st input parameter, such as MMHC, GSBN, F2SL_c, etc.

2nd input = data set

The data set is used for training. Columns are variables, and rows are observations. Support both continuous and discrete data sets. There are some simulated benchmark Bayesian network data sets in the data folder for using.

3rd input = type of data set

Since the data type is discrete or continuous, the 3rd input parameter is 'dis' or 'con'.

4th input = significance level

The significance level of hypothesis tests (e.g., G^2 test and Fisher's z-test), and is usually set to 0.01 or 0.05.

Outputs (two parameters):

1st output = learned global causal structure

An adjacency matrix of the learned global causal structure. The algorithm constructs undirected graph and attempts to direct all edges. If the element in the i^{th} row and j^{th} column of adjacency matrix is equal to 1, variable i is a parent of j.

2nd output = running time

Running time (in seconds) of an algorithm.

3.2 Local causal structure learning algorithms

Causal Learner has implemented 4 local causal structure learning algorithms, the full name of each algorithm is shown in Table 7, and then the references of all algorithms.

- ★ PCDbyPCD - PCD-by-PCD algorithm[8]
- ★ MBbyMB - MB-by-MB algorithm[9]
- ★ CMB - Causal Markov Blanket algorithm[10]
- ★ LCSFS - Local Causal Structure learning by Feature Selection (LCS-FS) algorithm[11]

Table 7: Full name of all local causal structure learning algorithms

[8] Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng. 2008. Partial orientation and local structural learning of causal networks for prediction. In *Causation and Prediction Challenge*. 93–105.

[9] Changzhang Wang, You Zhou, Qiang Zhao, and Zhi Geng. 2014. Discovering and orienting the edges connected to a target variable in a DAG via a sequential local learning approach. *Computational Statistics & Data Analysis* 77 (2014), 252–266.

[10] Tian Gao and Qiang Ji. 2015. Local causal discovery of direct causes and effects. In *Advances in Neural Information Processing Systems*. 2512–2520.

[11] Zhaolong Ling, Kui Yu, Hao Wang, Lei Li, and Xindong Wu. 2020. Using feature selection for local causal structure learning. *IEEE Trans. Emerg. Topics Comput. Intell.* DOI:10.1109/TETCI.2020.2978238 (2020).

The input and output parameter types and quantities are the same for all local causal structure learning algorithms as shown in Table 8.

Inputs (five parameters):	1st input = algorithm name
	2nd input = data set
	3rd input = type of data set (discrete or continuous)
	4th input = significance level
	5th input = target variable index
Outputs (five parameters):	1st output = learned target's parents
	2nd output = learned target's children
	3rd output = learned target's PC
	4th output = number of conditional independence tests
	5th output = running time

Table 8: Inputs and outputs of local structure learning algorithms

Inputs (five parameters):

1st input = algorithm name

Users can use one of the algorithm names in Table 7 as the 1st input parameter, such as PCD_by_PCD, MB_by_MB, CMB, etc.

2nd input = data set

The data set is used for training. Columns are variables, and rows are observations. Support both continuous and discrete data sets. There are some benchmark BN data sets in the data folder for using.

3rd input = type of data set

Since the data type is discrete or continuous, the 3rd input parameter is 'dis' or 'con'.

4th input = significance level

The significance level of hypothesis tests (e.g., G^2 test and Fisher's z-test), and is usually set to 0.01 or 0.05.

5th input = target variable index

Index of the target variable that the user is interested in. Our goal will be to find local causal structure of this variable.

Outputs (five parameters):

1st output = learned target's parents

A vector with the indexes of variables in the set of learned target's parents.

2nd output = learned target's children

A vector with the indexes of variables in the set of learned target's children.

3rd output = learned target's PC

A vector with the indexes of variables in the set of the union of the learned target's parents and children.

4st output = number of conditional independence tests

Number of conditional independence tests of an algorithm.

5st output = running time

Running time (in seconds) of an algorithm.

3.3 Markov blanket learning algorithms

Causal Learner has implemented 15 Markov blanket learning algorithms, the full name of each algorithm is shown in Table 9, and then the references of all algorithms.

- ★ GS - Grow/Shrink algorithm[12]
- ★ IAMB - Incremental Association-Based Markov Blanket (IAMB) algorithm[13]
- ★ InterIAMB - Inter-IAMB algorithm[14]
- ★ IAMBnPC - IAMBnPC algorithm[14]
- ★ InterIAMBnPC - Inter-IAMBnPC algorithm[14]
- ★ FastIAMB - Fast-IAMB algorithm[15]
- ★ FBED - Forward-Backward selection with Early Dropping algorithm[16]
- ★ MMMB - Min-Max MB algorithm[17]
- ★ HITONMB - HITON-MB algorithm[18]
- ★ PCMB - Parents and children based MB algorithm[19]
- ★ IPCMB - Iterative Parent-Child based search of MB algorithm[20]
- ★ MBOR - MB search using the OR condition algorithm[21]
- ★ STMB - Simultaneous MB discovery algorithm[22]
- ★ BAMB - Balanced MB discovery algorithm[23]
- ★ EEMB - Efficient and Effective MB algorithm[24]
- ★ MBFS - MB by Feature Selection algorithm[25]

Table 9: Full name of all Markov blanket learning algorithms

[12] Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. *In Advances in neural information processing systems*, pages 505–511, 2000.

[13] Ioannis Tsamardinos and Constantin Aliferis. 2003. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*. Citeseer.

[14] Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. Algorithms for large scale markov blanket discovery. In FLAIRS conference, volume 2, pages 376–380, 2003b.

[15] Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 4–9.

[16] Giorgos Borboudakis and Ioannis Tsamardinos. 2019. Forward-backward selection with early dropping. *J. Mach. Learn. Res.* 20, 1 (2019), 276–314.

[17] Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander Statnikov. 2003. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, 673–678.

[18] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classifification part i: Algorithms and empirical evaluation. Journal of Machine Learning Research, 11:171–234, 2010a.

[19] Jose M. Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. 2007. Towards scalable and data efficient learning of Markov boundaries. *Int. J. Approx. Reas.* 45, 2 (2007), 211–232.

[20] Shunkai Fu and Michel C. Desmarais. 2008. Fast Markov blanket discovery algorithm via local learning within single pass. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 96–107.

[21] Sergio Rodrigues De Morais and Alex Aussem. 2008. A novel scalable and data efficient feature subset selection algorithm. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'08)*. Springer, 298–312.

[22] Tian Gao and Qiang Ji. 2017. Efficient Markov blanket discovery and its application. *IEEE Trans. Cyber.* 47, 5 (2017), 1169–1179.

[23] Zhaolong Ling, Kui Yu, Hao Wang, Lin Liu, Wei Ding, and Xindong Wu. 2019. BAMB: A balanced Markov blanket discovery approach to feature selection. *ACM Trans. Intell. Syst. Technol.* 10, 5 (2019), 1–25.

[24] Hao Wang, Zhaolong Ling, Kui Yu, and Xindong Wu. 2020. Towards efficient and effective discovery of Markov blankets for feature selection. *Inf. Sci.* 509 (2020), 227–242.

[25] Zhaolong Ling, Kui Yu, Hao Wang, Lei Li, and Xindong Wu. 2020. Using feature selection for local causal structure learning. *IEEE Trans. Emerg. Topics Comput. Intell.* DOI:10.1109/TETCI.2020.2978238 (2020).

The input and output parameter types and quantities are the same for all Markov blanket learning algorithms as shown in Table 10.

Inputs (five parameters):	1st input = algorithm name
	2nd input = data set
	3rd input = type of data set (discrete or continuous)
	4th input = significance level
	5th input = target variable index
Outputs (three parameters):	1st output = learned target's MB
	2nd output = number of conditional independence tests
	3rd output = running time

Table 10: Inputs and outputs of Markov blanket algorithms

Inputs (five parameters):

1st input = algorithm name

Users can use one of the algorithm names in Table 9 as the 1st input parameter, such as MMMB, HITONMB, IAMB, etc.

2nd input = data set

The data set is used for training. Columns are variables, and rows are observations. Support both continuous and discrete data sets. There are some benchmark Bayesian network data sets in the data folder for using.

3rd input = type of data set

Since the data type is discrete or continuous, the 3rd input parameter is 'dis' or 'con'.

4th input = significance level

The significance level of hypothesis tests (e.g., G^2 test and Fisher's z-test), and is usually set to 0.01 or 0.05.

5th input = target variable index

Index of the target variable that the user is interested in. Our goal will be to find Markov blanket of this variable.

Outputs (five parameters):

1st output = learned target's MB

A vector with the indexes of variables in the set of learned target's Markov blanket by an Markov blanket learning algorithm.

2st output = number of conditional independence tests

Number of conditional independence tests of an algorithm.

3st output = running time

Running time (in seconds) of an algorithm.

4. Evaluation

Causal Learner provides abundant metrics for evaluating causal structure and MB learning algorithms (written in MATLAB). In terms of accuracy, global and local causal structure learning algorithms are evaluated using the same 7 metrics: Ar_F1 (Ar denotes arrow), Ar_precision, Ar_recall, SHD, Miss, Extra, and Reverse. MB learning algorithms are evaluated using 3 metrics: Adj_F1 (Adj denotes adjacent), Adj_precision, and Adj_recall. In terms of efficiency, global causal structure learning algorithms are evaluated using running time, while both local causal structure and MB learning algorithms are evaluated using running time and the number of conditional independence tests.

Since the efficiency metrics can be obtained directly after running the algorithm, it will not be described here. As for accuracy metrics, after the algorithm runs, Causal Learner will call the MATLAB file evaluation_GCS.m, evaluation_LCS.m, or evaluation_MB.m to compare the results learned by the algorithm with the correct results, and then calculate the corresponding accuracy metrics. Among them, global causal structure learning and local causal structure learning algorithms need to orient the edges, so their accuracy evaluation metrics will involve the accuracy of the direction. MB learning algorithms learn the set of MB, so the accuracy evaluation metrics of the MB learning algorithms do not involve the correctness of the direction.

4.1 Evaluating global causal structure learning algorithms

The input and output parameter types and quantities are the same for evaluating global causal structure learning algorithms (using evaluation_GCS.m) as shown in Table 11.

Inputs (two parameters):	1st input = learned global causal structure
	2nd input = graph
Outputs (seven parameters):	1st output = Ar_F1
	2nd output = Ar_precision
	3rd output = Ar_recall
	4th output = SHD
	5th output = Miss
	6th output = Extra
	7th output = Reverse

Table 11: Inputs and outputs of evaluating global causal structure learning algorithms

Inputs (two parameters):

1st input = learned global causal structure

An adjacency matrix of the learned global causal structure. The algorithm constructs undirected graph and attempts to direct all edges. If the element in the i^{th} row and j^{th} column of adjacency matrix is equal to 1, variable i is a parent of j.

2nd input = graph

An adjacency matrix of the true Bayesian network structure.

Outputs (seven parameters):

1st output = Ar_F1

$Ar\_F1 = 2 * (Ar\_Precision * Ar\_Recall) / (Ar\_Precision + Ar\_Recall)$.

The Ar_F1 score is the harmonic average of the Ar_precision and Ar_recall, where Ar_F1 = 1 is the best case (perfect Ar_precision and Ar_recall) while Ar_F1 = 0 is the worst case.

2st output = Ar_precision

Ar_precision = #correctly predicted arrowheads / #predicted arrowheads

The number of correctly predicted arrowheads in the output divided by the number of edges in the output of an algorithm.

3st output = Ar_recall

Ar_recall = #correctly predicted arrowheads / #true arrowheads

The number of correctly predicted arrowheads in the output divided by the number of true arrowheads in the true Bayesian network structure.

4st output = SHD

Structural Hamming Distance, the sum of the values of Miss, Extra, and Reverse as follows. Compared to Ar_F1, Ar_F1 considers not only extraneous edges, but also correct edges, while SHD only considers extraneous edges.

5th output = Miss

The number of missing edges in the global causal structure learned by the algorithm against the true Bayesian network structure.

6th output = Extra

The number of extra edges in the learned global causal structure.

7th output = Reverse

The number of edges with wrong directions according to the true Bayesian network structure.

4.2 Evaluating local causal structure learning algorithms

The input and output parameter types and quantities are the same for evaluating local causal structure learning algorithms (using evaluation_LCS.m) as shown in Table 12.

Inputs (five parameters):	1st input = learned target's parents
	2nd input = learned target's children
	3rd input = learned target's undirected
	4th input = target
	5th input = graph
Outputs (eight parameters):	1st output = Ar_F1
	2nd output = Ar_precision
	3rd output = Ar_recall
	4th output = SHD
	5th output = Miss
	6th output = Extra
	7th output = Reverse
	8th output = Undirected

Table 12: Inputs and outputs of evaluating local causal structure learning algorithms

Inputs (two parameters):

1st input = learned target's parents
A vector with the indexes of variables in the set of target's parents learned by a local causal structure learning algorithm.

2nd input = learned target's children
A vector with the indexes of variables in the set of target's children learned by a local causal structure learning algorithm.

3rd output = learned target's undirected
A vector with the indexes of variables in the set of target's PC learned by a local causal structure learning algorithm, which cannot be distinguished into parents or children.

4th input = target variable index
Index of the target variable that the user is interested in.

5th input = graph

An adjacency matrix of the true Bayesian network structure, and it is used for comparing the local causal structure around a target node learned by a local causal structure learning algorithm with the local causal structure around the target in the true Bayesian network structure.

Outputs (eight parameters):

1st output = Ar_F1

$Ar\_F1 = 2 * (Ar\_Precision * Ar\_Recall) / (Ar\_Precision + Ar\_Recall)$.

The Ar_F1 score is the harmonic average of the Ar_precision and Ar_recall, where Ar_F1 = 1 is the best case (perfect Ar_precision and Ar_recall) while Ar_F1 = 0 is the worst case.

2st output = Ar_precision

Ar_precision =#correctly predicted arrowheads / #predicted arrowheads

The number of correctly predicted arrowheads in the output divided by the number of edges in the output of an algorithm.

3st output = Ar_recall

Ar_recall = #correctly predicted arrowheads / #true arrowheads

The number of correctly predicted arrowheads in the output divided by the number of true arrowheads in the true local Bayesian network structure.

4st output = SHD

Structural Hamming Distance, the sum of the values of Miss, Extra, and Reverse as follows. Compared to Ar_F1, Ar_F1 considers not only extraneous edges, but also correct edges, while SHD only considers extraneous edges.

5th output = Miss

The number of missing edges in the local causal structure learned by the algorithm against the true local Bayesian network structure.

6th output = Extra

The number of extra edges in the learned local causal structure.

7th output = Reverse

The number of edges with wrong directions according to the true local Bayesian network structure.

8th output = Undirected

The number of edges that cannot be oriented in the learned local causal structure.

4.3 Evaluating Markov blanket learning algorithms

The input and output parameter types and quantities are the same for evaluating Markov blanket learning algorithms (using evaluation_MB.m) as shown in Table 13.

Inputs (three parameters):	1st input = learned target's MB
	2nd input = target variable index
	3rd input = graph
Outputs (three parameters):	1st output = Adj_F1
	2nd output = Adj_precision
	3rd output = Adj_recall

Table 13: Inputs and outputs of evaluating Markov blanket learning algorithms

Inputs (three parameters):

1st input = learned target's MB
A vector with the indexes of variables in the set of learned target's Markov blanket.

2nd input = target variable index
Index of the target variable that the user is interested in.

3rd input = graph
An adjacency matrix of the true Bayesian network structure, and it is used for comparing the Markov blanket around a target node learned by a Markov blanket learning algorithm with the Markov blanket around the target in the true Bayesian network structure.

Outputs (three parameters):

1st output = Adj_F1
$Adj\_F1 = 2 * (Adj\_Precision * Adj\_Recall) / (Adj\_Precision + Adj\_Recall)$.
The Adj_F1 score is the harmonic average of the Adj_precision and Adj_recall, where Adj_F1 = 1 is the best case (perfect Adj_precision and Adj_recall) while Adj_F1 = 0 is the worst case.

2st output = Adj_precision

Adj_precision = #correctly predicted nodes /#predicted nodes.

Adj_precision is the number of true positives in the learned Markov blanket divided by the number of the learned Markov blanket.

3st output = Adj_recall

Adj_recall = #correctly predicted nodes / #true nodes.

Adj_recall is the number of true positives in the learned Markov blanket divided by the number of the true Markov blanket.