AFRL-RI-RS-TR-2021-197

LOCAL STRUCTURE LEARNING AND KNOWLEDGE AUGMENTATION LEARNING

RENSSELAER POLYTECHNIC INSTITUTE

NOVEMBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-197 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
GENNADY R. STASKEVICH
Work Unit Manager

/ S /
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
NOVEMBER 2021	FINAL TECHNICAL REPORT	MARCH 2017	MAY 2021

4. TITLE AND SUBTITLE

LOCAL STRUCTURE LEARNING AND KNOWLEDGE AUGMENTATION LEARNING

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-17-2-0132	N/A	62702E
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
D3MP	S0	07

6. AUTHOR(S)

Qiang Ji

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Rensselaer Polytechnic Institute
110 8th street
Troy NY 12180

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Air Force Research Laboratory/RISC 525 Brooks Road Rome NY 13441-4505	RI	AFRL-RI-RS-TR-2021-197

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Local structure learning is to discover the local structure for a target variable. A key concept for local structure is the Markov Blanket (MB). The MB of a target variable consists of those variables, which jointly contain all the information needed to predict the behaviors of the target variable. MB learning is to discover the MB of the target node. The research objectives are: 1) development of robust and efficient methods for MB learning; 2) application of the MB learning to various machine learning tasks, including structured feature selection, classification, and causal structure learning, and 3) incorporation of the MB learning methods into D3M environment and evaluation of their performance on D3M datasets.

15. SUBJECT TERMS

Markov blanket learning, probabilistic graphical model, causal network learning, structured feature selection and classification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	38

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
GENNADY R. STASKEVICH	N/A

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1. SUMMARY

The goal of this project is to develop local structure learning methods to distill local structural information from raw data and apply the distilled structure to different machine learning tasks. Local structure learning is to discover the structural relationships among variables with respect to a target variable. A key concept for local structure learning is the Markov Blanket (MB). The MB of a target variable consists of those variables, which jointly contain all the information needed to predict the behaviors of the target variable. Probabilistically, given its MB, a target variable is independent of all other variables. MB learning is to learn from data a graph that represents the MB of the target node. The objectives of this research are: 1) development of robust and efficient methods for MB learning under different conditions; 2) application of MB learning to various machine learning (ML) tasks, including feature selection, classification, and causal structure learning, and 3) implementation of our MB learning methods into D3M environment and evaluation of their performance for different ML tasks on D3M datasets.

2. INTRODUCTION

Structured learning and prediction is an important area of research in machine learning. Conventional machine learning techniques usually learn a mapping function that maps input variables to the output variables. Even with good prediction performance, conventional ML methods cannot explain their prediction and cannot effectively model the underlying data generation mechanism. In contrast, structured learning deals with the problem of learning the structured relationships among input and output variables and leverage both the values of the variables and their relationships to jointly perform the prediction. Moreover, as the structured relationships between input and output variables capture the inherent and intrinsic data dependencies, structured learning and prediction are more stable, generalizable across datasets, and are more explainable. Structured learning has applications in many fields. For example, in computer vision, each image pixel contains information of a small part of objects. Pixels' interaction and layout follow the structure of the objects. In molecular biology, the genes or proteins follow a rigid molecular structure to bond together by chemical properties.

Probabilistic graphical model (PGM) is a dominant approach for structured learning. By treating each variable in the problem domain as a node in a graph and the interactions among these variables as graph links and with their built-in local conditional independence, PGMs can compactly capture the joint probabilistic distribution of random variables. By capturing the interactions among input and output variables, PGMs allow not only predicting the output variables given input variables but also discovering the most relevant input variables for a certain output variable. In particular, the directed PGMs such as Bayesian Networks (BNs) employ directed edges to capture the unilateral interactions (often causation) among random variables. Using such a structure, BNs have been successfully applied to a wide variety of fields, including bioinformatics, natural language processing, and computer vision.

Structured learning and prediction with PGMs starts with model learning, including both the model (graph) structure and parameters, from training data. Methods for learning the model can be divided into global and local approach. Global approach learns the entire model over all variables simultaneously. While allowing to capture the interactions among all nodes, the global

approach cannot scale up well as the model space increases exponentially with the number of nodes. Many structural assumptions, such as tree structure, are often made to simplify the learning, yielding inaccurate graph structures. The global structure learning difficulty can be reduced by decomposing a large graph into smaller sub-graphs so that each sub-graph can be learned separately. One such approach is the local to global method, whereby the local structure for each node is learned separately and the global graph can be obtained by combining the local structures after resolving any conflicts. Such an approach can significantly improve the model learning efficiency, with minimum or no loss in structural accuracy. Moreover, for many machine learning problems, the global graph is often unnecessary as we are only interested in predicting certain target (e.g., label) variables. Through this research, we developed different local structure learning methods under varying operating conditions, apply them to several machine learning problems, and demonstrated their performance on benchmark and D3M datasets.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Simultaneous Markov Blanket Learning

In this section, we first start with preliminaries that are necessary for subsequent discussion on various MB learning methods. We then introduce an efficient method for MB learning. This section ends with an empirical evaluation of the proposed MB learning method on benchmark datasets against state of the art (SOTA) MB learning methods.

Consider a set of random variables $\boldsymbol{V}$, its joint distribution can be captured by a Bayesian Network (BN). Specifically, a BN for $\boldsymbol{V}$ consists of the structure $\mathcal{G}$ and the corresponding probability parameters $\boldsymbol{\theta}$. The structure $\mathcal{G}$ is a directed acyclic graph (DAG) with each node corresponding to a random variable in $\boldsymbol{V}$. A directed link between two connected nodes captures the directed (such as causal) dependencies between two corresponding random variables. If there is a directed link pointing to node Y from node X, X is a parent of Y and Y is a child of X. Two non-adjacent nodes sharing the same child are spouse. The probability parameters $\boldsymbol{\theta}$ represent the conditional probability distribution (CPD) of each node $X \in \boldsymbol{V}$ given its parents.

In a BN, a Markov Blanket of a target variable T consists of the target node's parents, children, and spouses. We denote the Markov Blanket of node T as $\boldsymbol{B}_T$. For example, as shown in Figure 1, the Markov Blanket of target node T consists of colored nodes $\{P, S, C\}$, where node P is the parent, node C is the child and node S is the spouse. An important property of the Markov Blanket is that $\boldsymbol{MB}_T$ is the minimal set of nodes conditioned on which all other nodes are independent of T, i.e.,

$$X \perp\!\!\!\perp T \mid \boldsymbol{MB}_T, \qquad \forall X \in \{\boldsymbol{V} \setminus T \setminus \boldsymbol{MB}_T\}$$

Markov Blanket learning algorithms are aimed at estimating the MB of a target node from a set of $i.i.d.$ observed samples following an unknown distribution $\mathcal{P}$. There are mainly two categories of MB learning algorithms: score-based and constraint-based. Our proposed method belongs to the constraint-based approach, where MB is learned by employing independence

tests. Under the *faithfulness condition*, the MB of a target node is uniquely identifiable, where the *faithfulness condition* states that a DAG $\mathcal{G}$ is faithful to $\mathcal{P}$ if all and only the conditional independencies true in $\mathcal{P}$ are entailed by $\mathcal{G}$.

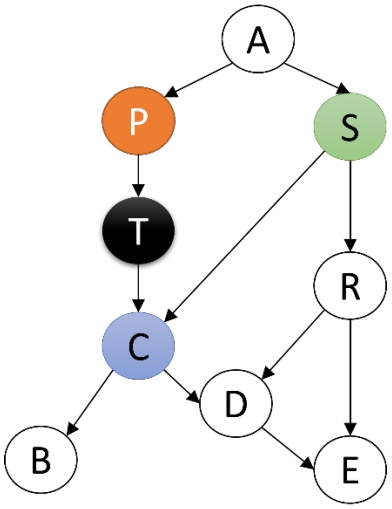

Figure 1. Example of Markov Blanket

The overall framework of the constraint-based MB learning algorithms, including Parents Children based Markov Blanket (PCMB) [1] and Iterative Parent-Child based search of MB (IPCMB) [2], consists of three steps: 1) find the parents and children (PC) set of the target node via an exhaustive search; 2) enforce the *symmetry constraint* to remove false positive nodes in the PC set, and 3) find the spouses to complete the MB. The *symmetry constraint* in Step 2 states that for a PC node X to be valid, the target node T must also be in the PC set of node X, i.e., $X \in \boldsymbol{PC}_T$ and $T \in \boldsymbol{PC}_X$.

While the complexity of MB learning methods is mainly resulted from step 1 due to it's need for exhaustive search for the PC set, step 2 further increases their complexity by $|\boldsymbol{PC}_T|$ times, where $|\boldsymbol{PC}_T|$ denotes the size of the PC set of the target node. To improve MB learning efficiency, we propose Simultaneous Markov Blanket (STMB) [3] to efficiently find the MB of a target node. STMB is developed based on our discovery of the co-existence theorem between the false positive PC nodes and the spouse nodes:

Co-existence Property Theorem: Given $\boldsymbol{PC}_T$ that is the identified PC set for target node T, the only false positive nodes in $\boldsymbol{PC}_T$ are descendants of T, due to an unblocked path from T to false positive PC nodes via T 's spouses.

By exploiting the co-existence property, STMB allows simultaneously identifying the false PCs and find the spouse nodes, hence eliminating the symmetry check step and significantly reducing the MB learning complexity without any loss in accuracy. Algorithm 1 summarizes the two steps in STMB: 1) identify the parents and children (PC) set, which is the same as existing MB learning methods; and 2) find the spouse and remove false positive PC nodes

Algorithm 1 STMB Algorithm

Require: Data $\mathcal{D}$; target node T

1: $\mathbf{CanMB}_T \leftarrow \boldsymbol{V} \setminus T$;
{Step 1: find the PC set}
2: $[\mathbf{PC}_T, \mathbf{Sep}_T] \leftarrow$ RecogPC(T, $\mathbf{CanMB}_T$, $\mathcal{D}$)
{Step 2: find spouses and remove descendants}
3: $\mathbf{pouse}_T \leftarrow \emptyset$; $\mathbf{remove} \leftarrow \emptyset$
4: **for** each Y $\in \mathbf{PC}_T$ **do**
5: **for** each $X \in \{\text{CanMB}_T \setminus \mathbf{PC}_T\}$ **do**
6: **if** $X \; !\perp\!\!\!\perp \; T \mid \{\mathbf{Sep}_T\{X\} \cup Y\}$ **then**
$\perp\!\!\!\perp$

9: **break**;
10: **else**
11: $\mathbf{spouse}_T\{Y\} \leftarrow \mathbf{spouse}_T\{Y\} \cup X$;
12: **end if**
13: **end if**
14: **end for**
15: **end for**
16: $\mathbf{PC}_T \leftarrow \mathbf{PC}_T \setminus \mathbf{remove}$;
17: **for** each Y in $\mathbf{spouse}_T$ **do**
18: **for** each S in nonempty $\mathbf{spouse}_T\{Y\}$ **do**
19: $\mathbf{testSet} \leftarrow \mathbf{PC}_T \cup \mathbf{spouse}_T\{Y\} \setminus S$;
20: **if** $S \perp\!\!\!\perp T \mid \mathbf{testSet}$ **then**

22: **end if**
23: **end for**
24: **end for**
25: $\leftarrow \mathbf{PC}_T$;
26: **for** each $X \in \boldsymbol{M}$ **do**
27: **if** $X \perp\!\!\!\perp T \mid \{\mathbf{PC}_T \cup \mathbf{spouse}_T \setminus X\}$ **then**

29: **end if**
30: **end for**
31: $\mathbf{MB} \leftarrow \mathbf{spouse}_T \cup \mathbf{PC}_T$;

simultaneously. Specifically in Step 2, STMB first looks for a node $Y \in \boldsymbol{PC}_T$ that unblocks one path from target node T to a candidate spouse node X, i.e., $X \in \boldsymbol{V} \setminus \boldsymbol{PC}_T$. If such an Y exists, X is a spouse (Line 11) and a further test (Line 7) is then performed to determine if Y is a false positive PC or a legitimate child of the target node. The process repeats to enumerate each PC node (Line 4) and each candidate spouse node (Line 5). Step 2 not only identifies the spouse nodes but also their children. By identifying spouses and removing false positive PC nodes simultaneously,

STMB improves the MB learning efficiency significantly. Compared to the existing MB learning algorithm IPCMB [3], STMB reduces the worst-time complexity of the IPCMB by $\mathcal{O}(C)$, with C being the largest size of the PC sets of all the nodes in the graph. For larger and more densely connected networks, STMB can achieve significant speedups. Furthermore, STMB remains an exact method. Under the faithfulness condition, STMB is both sound and complete, i.e., it finds all and only the Markov Blanket nodes of the target node. Further details on STMB's optimal properties and their proofs can be found in [3].

To empirically demonstrate the accuracy and efficiency of STMB, we evaluate STMB on three benchmark Markov Blanket learning datasets[1]: CHILD, ALARM, HAILFINDER. Their statistical information is summarized in Table 1. We compare STMB against the state of the art IPCMB algorithm. To measure the accuracy of MB learning, we employ the MB discovery error metric, namely the distance between true MB and estimated MB: $d = \sqrt{(1 - \text{Precision})^2 + (1 - \text{Recall})^2}$. Precision is the number of true positives in the detected MB divided by the total size of the detected MB set. Recall is the number of true positives in the detected MB divided by the size of the ground truth MB. The lower the distance d, the better the accuracy of MB learning. We run the MB learning algorithm for each node in each network. We repeat 10 runs with different sample sets and report the average distance. To measure the MB learning efficiency, we employ the average number of conducted independence tests as the efficiency metric. For all algorithms, we employ standard mutual information based independence test and the threshold is set to be 0.02.

Table 1. Statistical Information of the Benchmark Datasets

Dataset	#Variables	#Edges	Maximum #States
CHILD	20	25	6
ALARM	37	46	4
HAILFINDER	56	66	11

Results are summarized in Table 2. On all three datasets, STMB achieves comparable accuracy to IPCMB. But computationally, STMB is significantly more efficient than IPCMB. For example, on the ALARM dataset, STMB is 4.3 times faster than IPCMB on average, and about 7 times faster with 500 samples. We can obtain similar results on two other datasets.

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

Table 2. Evaluation of STMB

Dataset	Size	Distance		#Independence Test	
		IPCMB	STMB	IPCMB	STMB
CHILD	500	0.57±0.06	0.63±0.06	434±50	83±7
	1000	0.39±0.04	0.42±0.05	387±42	89±4
	5000	0.24±0.05	0.18±0.03	283±21	83±2
	MEAN	**0.40**	0.41	368	**85**
ALARM	500	0.51±0.05	0.56±0.04	809±91	127±4
	1000	0.43±0.05	0.48±0.04	694±29	164±6
	5000	0.37±0.04	0.36±0.04	489±9	175±2
	MEAN	**0.44**	0.47	664	**155**
HAIL	500	1.08±0.03	1.24±0.02	2079±156	265±12
	1000	1.06±0.02	1.00±0.01	1493±143	250±7
	5000	1.10±0.01	1.04±0.01	308±37	159±4
	MEAN	**1.08**	1.09	1293	**225**

3.2 Robust MB Learning Under Insufficient Data

As the constraint-based MB learning methods employ the independence tests to discover the MB, they are hence susceptible to the reliability of the independence tests. Under insufficient data, independence tests, particularly high order independence tests, can be unreliable [4]. Independence test error, even one mistake at the early stage of the learning, can propagate throughout the learning process, causing a sequence of errors and resulting in an erroneous MB. To address this issue, we propose a robust MB learning method to allow accurate MB discovery even under insufficient data. Such a method is important since even in the era of big data, there are still domains in which the availability of data is very limited. For example, in biological or clinical disciplines, data can be severely insufficient either because of high cost or lack of cases from which data is collected [5]. Furthermore, even for the applications with vast amount of data, the data may not adequately cover all possible states of the variables, leading to insufficient data for certain states. For example, the observed data under absence of earthquake is adequate, while the observed data under occurrence of earthquake is limited, due to the fact that earthquake rarely happens in nature [6].

Specifically, we propose to introduce Bayesian approaches to improve both mutual information (MI) and hypothesis testing based independence tests. For MI based approach, we employ empirical Bayesian approach for better MI estimation under limited data. For hypothesis testing based approach, we reformulate the Bayes Factor as a well-defined statistical test.

3.2.1 Bayesian Mutual Information.

The mutual information of two discrete random variables X and Y is a measure of the amount of information shared between them and is defined as:

$$MI(X;Y) = \sum_{i=1}^{K_x} \sum_{j=1}^{K_y} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \quad (1)$$

Equation 1. Definition of Mutual Information

where K_x and K_y denote the total number of possible states of X and Y respectively. $P(x_i, y_j)$, $P(x_i)$, and $P(y_i)$ represent the joint probability of (X, Y), and the marginal probabilities of X and Y respectively. Ideally, $MI(X;Y) = 0$ if and only if X and Y are independent. In practice, the true MI is unknown, and the estimated $\widehat{MI}$ from data is always larger than zero. In the following, we denote the probability distribution parameters as $\boldsymbol{\theta}$, i.e., $\theta_i = P(x_i)$, $\theta_j = P(y_j)$ and $\theta_{ij} = P(x_i, y_j)$.

Conventionally, maximum likelihood estimation (MLE) method is employed to estimate distribution parameters from data $\mathcal{D}$ as $\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}\, p(\mathcal{D}|\boldsymbol{\theta})$. MLE asymptotically obtains the true parameter distribution. MLE, however, is unreliable when data is insufficient. To accurately estimate parameter $\boldsymbol{\theta}$ under limited data, we employ the Bayesian approach, i.e.,

$$\widehat{\boldsymbol{\theta}}^{Bayes} = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha}) d\boldsymbol{\theta} \quad (2)$$

Equation 2. Bayesian Estimation of Parameters

where $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha})$ is the posterior distribution of $\boldsymbol{\theta}$, given the training data $\mathcal{D}$ and the hyper-parameters $\boldsymbol{\alpha}$. $\boldsymbol{\alpha}$ specify the prior distribution of $\boldsymbol{\theta}$, which we assume follows the Dirichlet distribution for discrete random variables. As we have no prior preference on the elements of the Dirichlet distribution, we assume symmetric Dirichlet distribution, i.e., each entry in $\boldsymbol{\alpha}$ shares the same value and we denote it as α. Eq. 2 can be solved analytically with a closed-form solution $\hat{\theta}_i^{Bayes} = \frac{n_i + \alpha}{N + \alpha K}$, where K is the number of states for the random variable, n_i is the number of samples for state i, and $N = \sum_i^K n_i$. Hyper-parameter α is tuned empirically. We refer the mutual information estimated with $\widehat{\boldsymbol{\theta}}^{Bayes}$ as the pseudo-Bayesian MI since Bayesian estimation is only applied to the parameters instead of to the mutual information itself.

To obtain the Bayesian estimation of the mutual information, we propose to directly apply Bayesian method to MI estimation. The empirical Bayesian MI method [7] estimates the MI as follows

$$\widehat{\boldsymbol{MI}}^{eB} = \int \boldsymbol{MI}(X;Y|\boldsymbol{\theta}) \boldsymbol{p}(\boldsymbol{\theta}|\mathcal{D}, \alpha) d\boldsymbol{\theta} \quad (3)$$

Equation 3. Empirical Bayesian Estimation of Mutual Information

The method assumes the hyper-parameter α is known or is specified manually. Instead, we propose to learn α from data as $\alpha^* = \arg\max_\alpha p(\alpha|\mathcal{D})$, which maximizes the posterior of α. By assuming uniform distribution of $p(\alpha)$, we have $\alpha^* = \arg\max_\alpha p(\mathcal{D}|\alpha)$. The likelihood distribution $p(\mathcal{D}|\alpha)$ can be computed as,

$$p(\mathcal{D}|\alpha) = N!\frac{\Gamma(K\alpha)}{\Gamma(K\alpha+N)}\prod_{i=1}^{K}\frac{\Gamma(\alpha+n_i)}{\Gamma(\alpha)n_i!} \quad (4)$$

Equation 4. Likelihood Distribution $p(\mathcal{D}|\alpha)$

where $\Gamma(x)$ is the gamma function. $p(\mathcal{D}|\alpha)$ follows Polya distribution. We solve for α^* with a fixed-point update [8]. Given α^*, $\widehat{\boldsymbol{MI}}^{eB}$ can be computed using the empirical Bayesian MI (Eq. 3) with a closed-form solution:

$$\widehat{\boldsymbol{MI}}^{\boldsymbol{eB}} = \psi(N+\alpha^*K+1) - \sum_{i,j}\frac{n_{ij}+\alpha^*}{N+\alpha^*K}[\psi(n_i+\alpha^*K_y+1)+\psi(n_j+\alpha^*K_x+1) - \psi(n_{ij}+\alpha^*+1)] \quad (5)$$

Equation 5. The Closed-form Solution to Empirical Bayesian Mutual Information

where $\psi(x)$ is the digamma function. n_i and n_j are the number of samples for $X = i$ and $Y = j$ respectively, and n_{ij} is the number of samples for $(X, Y) = (i, j)$. Given the estimated MI, we compare it against a pre-defined threshold for independence test. Two random variables will be declared independent if the MI estimate is smaller than the threshold and dependent otherwise.

3.2.2 Bayesian Hypothesis Testing. We now introduce our method to improve the hypothesis testing based independence test. We first consider a widely used independence test, G test, which is derived from the likelihood ratio test with null hypothesis assuming two random variables are independent. g-statistic is calculated as

$$g = -2\sum_{i=1}^{K_x}\sum_{j=1}^{K_y} n_{ij}\ln\frac{\widehat{\theta}_i\widehat{\theta}_j}{\widehat{\theta}_{ij}} \quad (6)$$

Equation 6. Definition of the g-statistic

where distribution parameters $\widehat{\boldsymbol{\theta}}$ are obtained via MLE. g is assumed to follow a χ^2 distribution, based on which a hypothesis testing can be performed. As MLE is not reliable under insufficient data, Bayes Factor [9] (BF) was introduced to consider the ratio of the expected likelihoods of the null hypothesis (H_0) and the alternative hypothesis (H_1) over the posterior parameter distributions under the null and alternate hypothesis respectively, i.e.,

$$BF = \frac{P(\mathcal{D}|H_0, \alpha^0)}{P(\mathcal{D}|H_1, \alpha^1)} = \frac{\int P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta}|H_0, \alpha^0)\boldsymbol{d\theta}}{\int P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta}|H_1, \alpha^1)\boldsymbol{d\theta}} \quad (7)$$

Equation 7. Definition of the Bayes Factor

where hyper-parameters α^0 and α^1 respectively specify the symmetric Dirichlet prior distributions under null and alternative hypothesis, and they are tuned empirically. BF can be analytically computed with a closed-form solution [9]. To apply BF for independence test, the value of BF is compared to a pre-defined threshold η. If $BF > \eta$, the null hypothesis is accepted, i.e., two variables are independent. They are otherwise declared dependent. Instead of manually setting a threshold, we propose to formulate the Bayes Factor into a well-defined statistical test similar to the G test. To this goal, we firstly approximate Polya distribution $P(\mathcal{D}|\alpha)$ by multinomial

distribution, i.e.,

$$p(\mathcal{D}|\alpha) \approx p(\mathcal{D}|\tilde{\boldsymbol{\theta}}) = \frac{N!}{\prod_{i=1}^{K} n_i!} \prod_{i=1}^{K} \tilde{\theta}_k^{n_k} \quad (8)$$

Equation 8. Approximated Polya Distribution

where $\tilde{\theta}_k$ are the modified parameters of the multinomial distribution. $\tilde{\theta}_k = \frac{g(n_k,\alpha)}{g(N,K\alpha)}$ with $g(n_k, \alpha) = an_k + b\alpha$ and $\Lambda = \begin{pmatrix} a \\ b \end{pmatrix}$ are unknown coefficients. By plugging the $p(\mathcal{D}|\alpha)$ (defined in Eq. 4) into Eq. 8, it is clear that to satisfy Eq. 8, we must have $n_k \ln g(n_k, \alpha) = \ln\Gamma(\alpha + n_k) - \ln\Gamma(\alpha)$. Given $\{n_k\}_{k=1}^{K}$ and α, we can construct a system of K such equations, through which we can solve for Λ^* as

$$\Lambda^* = \arg\min_{\Lambda} ||M\Lambda - T||_2^2 \quad (9)$$

Equation 9. Estimation of Unknown Coefficients Λ

where $M = \begin{pmatrix} n_1, \alpha \\ n_2, \alpha \\ \dots \\ n_K, \alpha \end{pmatrix}$, $T = \begin{pmatrix} t(n_1, \alpha) \\ t(n_2, \alpha) \\ \dots \\ t(n_K, \alpha) \end{pmatrix}$, and $t(n_k, \alpha) = \exp(\frac{1}{n_k}(\ln\Gamma(\alpha + n_k) - \ln\Gamma(\alpha)))$.

Given $\Lambda^* = \begin{pmatrix} \mathrm{a}^* \\ \mathrm{b}^* \end{pmatrix}$, we have $\tilde{\theta}_k$ as $\tilde{\theta}_k = \frac{g(n_k,\alpha)}{g(N,K\alpha)} = \frac{\mathrm{a}^* n_k + \mathrm{b}^* \alpha}{\mathrm{a}^* N + \mathrm{b}^* K\alpha}$. $p(\mathcal{D}|\tilde{\boldsymbol{\theta}})$ can well approximate $p(\mathcal{D}|\alpha)$. We approximate the hypothesis likelihood under null and alternative hypothesis respectively and obtain a modified Bayes Factor $\widehat{BF}$ as

$$\widehat{BF} = \frac{P(\mathcal{D}|\tilde{\boldsymbol{\theta}}, H_0)}{P(\mathcal{D}|\tilde{\boldsymbol{\theta}}, H_1)} = \frac{\prod_{i=1}^{K_x} \tilde{\theta}_i^{n_i} \prod_{i=1}^{K_y} \tilde{\theta}_j^{n_j}}{\prod_{i=1,j=1}^{K_x K_y} \tilde{\theta}_{ij}^{n_{ij}}} \quad (10)$$

Equation 10. The Modified Bayes Factor $\widehat{BF}$

Following the construction of g-statistic in Eq. 6, the BF test statistic can then be computed as,

$$BF_{chi2} = -2\ln\widehat{BF} = -2\sum_{i=1}^{K_x}\sum_{j=1}^{K_y} n_{ij} \ln\frac{\tilde{\theta}_i \tilde{\theta}_j}{\tilde{\theta}_{ij}} \quad (11)$$

Equation 11. Definition of the BF_{chi2} Statistic

BF_{chi2} asymptomatically follows the χ^2 distribution. We set 5% as the default significance level. If p-value is smaller than the significance level, we reject the null hypothesis and accept the alternative hypothesis.

3.2.3 Bayesian Independence Test Evaluation. We empirically demonstrate the effectiveness of the proposed Bayesian independence tests in improving MB learning performance under limited data. We incorporate two Bayesian independence test methods into STMB. We denote sI^{eB} as the STMB with empirical Bayesian MI estimation and sBF_{chi2} as STMB with BF_{chi2} independence test. We follow the same experimental settings and employ the same evaluation metrics as in Section 3.1 to compare sI^{eB} and sBF_{chi2} against STMB, which performs independence tests using standard MI method. The results are summarized in Table 3.

From Table 3, we can see that on all datasets both sI^{eB} and sBF_{chi2} outperform STMB in terms of efficiency. Furthermore, accuracy on most of the datasets is also improved with Bayesian independence tests. On CHILD dataset with 300 samples, sBF_{chi2} improves distance by 0.35. HAILFINDER dataset is hard to learn because the number of node states is large. With limited samples, none of the three methods can perform well in terms of accuracy. But sI^{eB} and sBF_{chi2} significantly outperform STMB in efficiency, requiring much fewer independence tests. This evaluation demonstrates that the proposed Bayesian independence tests can improve both MB learning accuracy and efficient under insufficient data.

Table 3. Evaluation of STMB under Limited Data

		Distance			#Independence Test		
Dataset	Size	sI^{eB}	sBF_{chi2}	STMB	sI^{eB}	sBF_{chi2}	STMB
CHILD	80	0.78±0.10	0.70±0.08	0.84±0.00.1	49±4	51±3	588±69
	100	0.69±0.10	0.66±0.08	0.84±0.01	49±4	60±3	631±57
	300	0.50±0.10	0.47±0.08	0.82±0.01	58±3	110±8	456±29
	MEAN	0.66	**0.61**	0.83	**52**	74	558
ALARM	80	0.92±0.06	0.92±0.07	0.91±0.04	75±5	95±6	776±179
	100	0.85±0.07	0.79±0.05	0.91±0.03	75±5	105±8	827±148
	300	0.77±0.03	0.59±0.06	0.82±0.06	102±4	183±9	425±46
	MEAN	0.85	**0.77**	0.88	**84**	128	676
HAIL.	300	1.19±0.02	1.14±0.02	0.98±0.01	107±2	173±7	9310±461
	500	1.16±0.01	1.08±0.01	0.99±0.01	114±3	201±9	3758±294
	800	1.15±0.01	1.04±0.01	1.01±0.01	125±2	232±13	1384±96
	MEAN	1.17	1.09	**0.99**	**115**	202	4817

3.3 Causal Markov Blanket Learning

Learning the causality has been a fundamental task in many disciplines. The causality can not only help explain the underlying data generation mechanisms but can also lead to more stable and explainable predictions. Causal relations are usually represented in the form of a Directed Acyclic Graph (DAG). Causal discovery learns a DAG from observational data and can be performed globally or locally. In this section, based on MB discovery, we introduce a local causal discovery method [10] that learns the Causal Markov Blanket (CMB) of a target variable. Different from MB, CMB explicitly ascertains the direct causes and effects of the target variable.

Specifically, a Markov blanket (MB) cannot completely identify the causality w.r.t. the target variable as the parent and some child variables are indistinguishable. In fact, MB can only identify the causal identities of the variables that form the V-structure. As shown in Figure 2, since variables T, C, and S form a V-structure, variable C is a direct effect of target variable T and spouse variable S. However, merely from the MB, we fail to identify the cause-effect relationships between the target variable T and the PC set variables A, B, and D. In contrast, for a CMB, the causal relationships between the target variable and it's MB variables are ascertained as shown in Figure 3.

Existing local causal discovery methods generally aim to identify a subset of causal relationships among all the random variables. Local causal discovery (LCD) algorithm [11] and its variants [12, 13] find pairwise causal relations of two variables by performing independence tests between the two variables along with a designated ancestor variable. Bayesian Local Causal

Discovery (BLCD) [14] explores the Y-structure[2] among MB variables to infer the causal relationships. Both the LCD algorithm and the BLCD algorithm need to search among all the variables to identify the direct cause and effect variables for the target variable. To efficiently ascertain the causality w.r.t the target variable, we propose the Causal Markov Blanket (CMB) discovery method, which determines the exact causal identities of MB variables w.r.t a target variable by tracking their conditional independence changes, without finding the global causal structure.

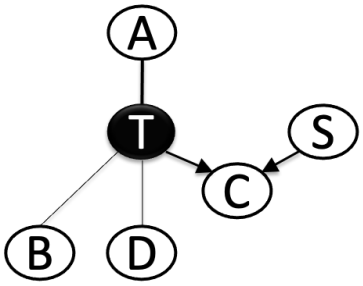

Figure 2. Example of Relations within a Markov Blanket

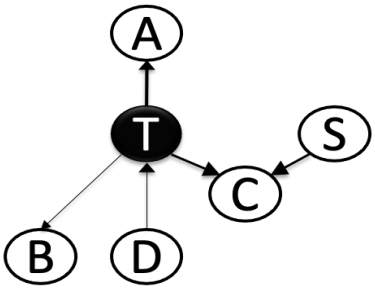

Figure 3. Example of Relations within a Causal Markov Blanket

The CMB discovery method is summarized in Algorithm 2. It has three major steps. In Step 1, CMB first performs MB discovery to identify the MB set for the target variable, including determining the causal identities for variables forming the V-structures. To ascertain the causal identities for the remaining PC variables, CMB then performs additional independence tests between every pair of PC variables, according to Lemma 1.

Lemma 1 For each pair of PC variables of target variable T, i.e., $(X, Y) \in \boldsymbol{PC}_T$, their independence relations fall into one of the four conditions:

C1. $X \perp\!\!\!\perp Y$ & $X \perp\!\!\!\perp Y \mid T$, this condition cannot happen.

C2. $X \perp\!\!\!\perp Y$ & $X \not\perp\!\!\!\perp Y \mid T \Longrightarrow X$ and Y are both the parents of T.

C3. $X \not\perp\!\!\!\perp Y$ & $X \perp\!\!\!\perp Y \mid T \Longrightarrow$ at least one of X and Y is a child of T.

[2] The Y structure is formed by four-variables, located in the vertices of a Y pattern.

C4. $X \not\perp\!\!\!\perp Y \;\&\; X \not\perp\!\!\!\perp Y \mid T \implies$ their identities are inconclusive and need further tests.

The independence relation between X and Y cannot satisfy **C1** because the path between X and Y is either blocked or unblocked by T. If independence relation between X and Y satisfies **C2**, then X and Y are both identified as parent variables (Line 4-5 in Algorithm 3). If independence relation between X and Y satisfies **C3**, X and Y can be a parent variable and a child variable or both are child variables. If one of X and Y is already identified as a parent variable, then the other one can be identified as a child variable (Line 7-12 in Algorithm 3). Otherwise, their identities are ambiguous and unidentifiable (Line 13-15 in Algorithm 3). If the independence relation between X and Y satisfies **C4**, it suggests that there may be another unblocked path from X to Y other than X - T - Y . The causal identities of X and Y are inconclusive (Line 18-19 in Algorithm 3).

In Step 2, CMB resolves the causal identities of the inconclusive variables from Step 1. For inconclusive variables pair (X, Y), it must examine variables beyond the PC set to determine whether there are other possible unblocked paths between X and Y , besides X - T - Y. To do so, CMB then finds the MB for variable X and performs the same independence tests between X and Y in Lemma 1, conditioned on the MB of X and the target variable T (Line 7-14 in Algorithm 2). Through Step 2, some inconclusive variables from Step 1 can be identified as parent or child variables. The remaining unidentifiable variables will be resolved in Step 3.

In Step 3, for every unidentifiable variable , CMB treats X as a target variable and repeats Steps 1 and 2. If the causal identity of T w.r.t to X is found (i.e., T is a child of X), the causal identity of X w.r.t to T is just it's reversal (i.e., X is a parent of T) (Line 17 - 19 in Algorithm 2). Step 3, however, cannot resolve the causal identities of all variables due to Markov equivalence. Hence, the causal identities of some variables will remain unidentified.

We evaluate the accuracy and efficiency of the CMB discovery algorithm on four medium to large benchmark datasets: ALARM, ALARM3, CHILD3, and INSURANCE3. The number of variables in those datasets ranges from 37 to 111. We use 1,000 data samples for all datasets. For comparison, we consider four global causal discovery methods PC [15], MMHC [16], GS [17], CS [18] and one local causal discovery method LCD [11]. For each global or local causal discovery algorithm, we find the global structure of all variables and then extract cause and effect variables w.r.t each variable. The CMB discovery directly finds the cause and effect variables w.r.t each variable. The cause and effect variables construct a local causal structure of the variable. To measure the accuracy of the discovered local causal structures w.r.t the ground-truth structure (and their Markov equivalents), we use the mean and standard deviation of their structural hamming distances (SHDs). To measure the efficiency, we report the mean and standard deviation of the number of conducted independence tests. The results are summarized in Table 4.

As shown in Table 4, in accuracy, CMB either outperforms other global or local causal discovery methods or achieves comparable results. With respect to efficiency, CMB improves the global causal discovery methods by one to two orders of magnitude. Compared to the local discovery method, CMB can also achieve more than one order of magnitude improvement. In conclusion, the CMB discovery obtains better or comparable accuracy, yet with greatly improved efficiency compared to other algorithms. Additional evaluations can be found in [10].

The performance of CMB depends on the accuracy of independence tests. Inaccurate independence tests, as a result of insufficient or imbalanced data, may introduce errors in each step of the CMB method, yielding erroneous causal identities. To address this issue, we are investigating methods to quantify the confidence level of an independence test, based on which

independence tests can be performed following the descending order of their confidence levels. This would reduce the influence of the independence test errors at the early stage on the subsequent learning process, hence yielding more robust and accurate CMB.

Algorithm 2 CMB Algorithm

INPUT: Data: $\mathcal{D}$; target variable: T.
OUTPUT: The causal identities of all variable w.r.t to target variable : ID_T ; The individual causal identity of a variable w.r.t T: id_T ; The causal identity of variable X w.r.t T: $ID_T(X)$.
{Step 1: Identify MB set for each variable and ascertain the causal identities for the PC variables}
Initial ID_T by assigning $id_T = 0$ for each variable;
$[\boldsymbol{MB}_T, \boldsymbol{PC}_T] \leftarrow$ FindMB$(T, \mathcal{D})$;
$\boldsymbol{Z} \leftarrow \emptyset$;
$ID_T \leftarrow$ CausalSearch$(\mathcal{D}, T, \boldsymbol{PC}_T, \boldsymbol{Z}, ID_T)$;
{Step 2: Further test the variables with $id_T = 4$}
for one $X \in$ each pair (X, Y) with $id_T = 4$ **do**
  $\boldsymbol{MB}_X \leftarrow$ FindMB$(X, \mathcal{D})$;
  $\boldsymbol{Z} \leftarrow \{\boldsymbol{MB}_X, \backslash T\} \backslash Y$;
  $ID_T \leftarrow$ CausalSearch$(\mathcal{D}, T, \boldsymbol{PC}_T, \boldsymbol{Z}, ID_T)$;
  if no element of ID_T is equal to 4 **then**
    break;
  end if
end for
$ID_T(X) \leftarrow 3, \forall X$ that $ID_T(X) = 4$;
{Step3: Resolve the variables with $id_T = 3$}
for each X with $id_T = 3$ **do**
  Find ID_X and update ID_T according to ID_X;
  if $ID_X(T) = 2$ **then**
    $ID_T(X) = 1$;
    for every Y in $id_T = 3$ variable pairs (X, Y) **do**
      $ID_T(Y) = 2$;
    end for
    if no element of ID_T is equal to 3 **then**
      break;
    end if
  end if
end for
Return: ID_T.

Algorithm 3 CausalSearch Subroutine

INPUT: Data: $\mathcal{D}$; target variable: T; the PC set of T: $\boldsymbol{PC}_T$; the conditioned variable set: $\boldsymbol{Z}$; the current causal identities of all variables w.r.t T : ID_T.
OUTPUT: The new causal identities of all variables w.r.t : ID_T.
{Step 1: Check independence conditions between variable pairs in PC set.}
for every $(X, Y) \in \boldsymbol{PC}_T$ **do**
  if $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ and $X \not\perp\!\!\!\perp Y|\boldsymbol{Z} \cup T$ **then**
    $ID_T(X) \leftarrow 1; ID_T(Y) \leftarrow 1;$
  else if $X \not\perp\!\!\!\perp Y|\boldsymbol{Z}$ and $X \perp\!\!\!\perp Y|\boldsymbol{Z} \cup T$ **then**
    if $ID_T(X) = 1$ **then**
      $ID_T(Y) \leftarrow 2;$
    end if
    if $ID_T(Y) = 1$ **then**
      $ID_T(X) \leftarrow 2;$
    end if
    if $ID_T(X) \neq 1$ and $ID_T(Y) \neq 1$ **then**
      $ID_T(X) \leftarrow 3; ID_T(Y) \leftarrow 3;$
    end if
    add (X, Y) to variable pairs with $id_T = 3$;
  else
    add (X, Y) to variable pairs with $id_T = 4$;
    $ID_T(X) \leftarrow 4; ID_T(Y) \leftarrow 4;$
  end if
end for
{Step 2: Ascertain identities for $id_T = 3$ variable pairs with known parents}
for every X such that $ID_T(X) = 1$ **do**
  for every Y in $id_T = 3$ variable pairs (X, Y) **do**
    $ID_T(Y) \leftarrow 2;$
  end for
end for
Return: ID_T.

Table 4. CMB Evaluation on Benchmark Datasets

Datasets	Methods	SHD	# Independent Test
ALARM	PC	4.10 ± 0.19	4.0*e*3 ± 4.0*e*2
	MMHC	3.46 ± 0.23	1.8*e*3 ± 1.7*e*3
	GS	2.39 ± 0.44	586.5 ± 72.2
	CS	**1.43 ± 0.10**	331.4 ± 61.9
	LCD2	2.49 ± 0.00	1.4*e*3 ± 0
	CMB	1.81 ± 0.11	**53.7 ± 4.5**
ALARM3	PC	7.30 ± 0.68	1.6*e*4 ± 4.0*e*2
	MMHC	5.53 ± 0.27	3.7*e*3 ± 6.1*e*2
	GS	3.64 ± 0.13	2.1*e*3 ± 1.2*e*2
	CS	**3.38 ± 0.13**	699.1 ± 60.4
	LCD2	3.85 ± 0.00	1.2*e*4 ± 0
	CMB	3.73 ± 0.11	**50.3 ± 6.2**
CHILD3	PC	7.76 ± 0.98	8.3*e*4 ± 2.9*e*3
	MMHC	4.00 ± 0.93	6.6*e*3 ± 8.2*e*2
	GS	4.57 ± 0.33	4.5*e*4 ± 2.2*e*3
	CS	4.37 ± 0.23	2.6*e*4 ± 3.9*e*3
	LCD2	5.03 ± 0.00	6.6*e*3 ± 0
	CMB	**2.36 ± 0.31**	**78.2 ± 15.2**
INSUR3	PC	8.55 ± 0.81	2.5*e*5 ± 1.2*e*4
	MMHC	5.68 ± 0.43	3.1*e*4 ± 5.2*e*2
	GS	4.57 ± 0.33	4.5*e*4 ± 2.2*e*3
	CS	4.37 ± 0.23	2.6*e*4 ± 3.9*e*3
	LCD2	5.03 ± 0.00	6.6*e*3 ± 0
	CMB	**4.30 ± 0.21**	**159.8 ± 38.5**

3.4 MB Learning Applications

Markov Blanket discovery for local structured learning has various applications. We focus on structured feature selection, global causal network learning, structured classification, and structured data imputation.

3.4.1 MB Learning for Structured Feature Selection. Feature selection is a dimensionality reduction technique. Roughly speaking, feature selection is to select a subset of features based on an evaluation measure. Feature selection is an important technique mainly due to: 1) it provides a concise and informative representation of the observed data; 2) it can reduce the complexity of the learning process; and 3) it can improve the accuracy of learning models by removing irrelevant features. Feature selection has been widely applied to different areas, and its effectiveness has been demonstrated in many practical applications [19, 20].

The majority of existing feature selection methods are aimed at selecting a subset of features to optimize the performance of a learning model. It is hence model-dependent. Furthermore, the

existing methods often fail to consider the structured relationships among features. Such structured relationships are important for accurate and robust prediction. We thus are motivated to capture and leverage the structural relations among features via MB learning for feature selection. By setting the class label as the target variable, feature selection with MB learning is formulated as finding feature variables that belong to the MB of the label variable.

Since MB learning only employs the independence relationships between features and target variable, MB-based feature selection is model (classifier) agnostic. In addition, under the faithfulness condition, structured feature selection with MB learning has the following optimal properties:

Maximal Mutual Information Theorem: Let $\boldsymbol{MB}$ be the Markov Blanket for a target T, and $\boldsymbol{V}$ be the entire feature set to the target T, then $\mathrm{MI}(\boldsymbol{MB}; T) \geq \mathrm{MI}(\boldsymbol{S}; T), \forall\, \boldsymbol{S} \subseteq \boldsymbol{V}$.

Minimum Feature Set Theorem: Let $\boldsymbol{MB}$ be the Markov Blanket for a target , and $\boldsymbol{V}$ be the entire feature set to the target T . Let $X \in \boldsymbol{V} \backslash \boldsymbol{MB}$ and $Y \in \boldsymbol{MB}$, then: a) $\mathrm{MI}(\boldsymbol{MB};\, T) = \mathrm{MI}(\boldsymbol{MB} \cup X; T)$; and b) $\mathrm{MI}(\boldsymbol{MB};\, T) > \mathrm{MI}(\boldsymbol{MB} \backslash Y;\, T)$

Minimal Bayes Error Theorem: Let $\boldsymbol{MB}$ be the Markov Blanket for a target T and $\boldsymbol{V}$ be the entire feature set to the target . The $\boldsymbol{MB}$ feature set minimizes both the lower bound and upper bound of the Bayes error.

The maximum mutual information theorem states that features selected with MB learning contain all the information about the target variable. The minimum feature set theorem states that the MB feature set is the smallest feature set that contains the maximum mutual information to the target variable. Finally, the minimal Bayes error theorem states that classification with the MB selected features yields the minimal Bayes classification error. Formal proofs for the three theorems can be found in [3].

To empirically demonstrate the effectiveness of structured feature selection with MB learning, we compare it's performance for object recognition on PASCAL Visual Object Classes (VOC 2007) dataset [21] against two existing feature selection methods: MI-IS [19] and PQ [22]. MI-IS [19] employs mutual information to perform feature selection, while PQ [22] performs feature selection through product quantization. Following the protocol in [19, 23], each image is represented by Fisher Vector (FV) features. Two metrics are used to evaluate the feature selection performance: feature compression ratio and classification accuracy. Compression ratio is calculated as the ratio of the size of original feature vector to the average size of selected features. It measures feature selection efficiency. Accuracy is measured by the mean Average Precision (mAP). The STMB method is employed to perform MB learning. For STMB, original feature set is broken down into segments of sizes K for efficient computation. We employ the linear SVM classifier for classification.

As shown in Table 5, STMB selects a smaller feature set at high compression ratio, and, in the meantime, achieves better accuracy compared to the other two methods. For example, STMB achieves mAP 54.4% with a compression ratio of 608, while MI-IS achieves mAP 52.70% with a compression ratio of 512 and PQ achieves mAP 54.0% with a compression ratio of 128. By considering underlying structured relationships among features, STMB can select a much smaller set of features that still contain the most information to the target variable, and thus achieves competitive accuracy with high compression ratio. Additional evaluations can be found in [3].

Table 5. Evaluation of Structured Feature Selection

Method	Compression Ratio	mAP(%)
STMB	1	59.40
	608, K=100	54.40
	704, K=200	54.00
MI-IS [19]	1	58.57
	256	56.82
	512	52.70
	1024	46.52
PQ [22]	1	58.80
	128	54.00
	256	50.30

To further evaluate the performance of the MB-based feature selection under insufficient data, we apply the Bayesian methods introduced in Section 3.2 to perform robust independence testing during MB learning. Specifically, we compare standard STMB to sI^{eB} and sBF_{chi2} (Bayesian STMB), where sI^{eB} denotes the STMB with empirical Bayesian MI estimation and sBF_{chi2} denotes the STMB with BF_{chi2} independence test. We consider five benchmark UCI classification datasets[3]: Breast, Congress, Heart, Parkinsons and Spect. Statistical information of these datasets is summarized in Table 6. K-Nearest Neighbor (KNN) classifier is employed and F1-score is applied as the metric of classification accuracy. The compression ratio is computed as $\frac{\#\text{entire features}}{\#\text{selected features}}$. sI^{eB} doesn't have results on the Breast dataset because no feature is selected.

Table 6. Statistical Information of UCI Datasets

Dataset	#Samples	#Features
Breast	699	9
Congress	435	16
Heart	303	75
Parkinsons	195	23
Spect	267	22

From Table 7, we can see that sBF_{chi2} is always better than STMB in terms of accuracy. Both sI^{eB} and sBF_{chi2} achieve higher compression ratio than STMB. For example, on Heart dataset, compared to STMB, sBF_{chi2} achieves 6.0% accuracy improvement with 4.62 times improvement in compression ratio. With Bayesian independence test, features that are most relevant to the class label are selected and as a result, classification performance is improved. Additional evaluations on standard STMB version Bayesian STMB for feature selection can be found in Table 12.

[3] http://archive.ics.uci.edu/ml/datasets.php

Table 7. Evaluation of Structured Feature Selection under Insufficient Data

Dataset	F1-score/Compression Ratio		
	sI^{eB}	sBF_{chi2}	STMB
Breast	–/–	0.92/**2.70**	**0.93**/1.23
Congress	**0.95**/**16.67**	**0.95**/**16.67**	0.94/1.79
Heart	0.77/**12.50**	**0.86**/4.76	0.80/1.03
Parkinsons	0.59/**25.00**	**0.83**/2.63	0.79/1.05
Spect	0.72/**1.47**	**0.74**/1.37	0.71/1.35

3.4.2 MB Learning for Global Causal Discovery. The global causal discovery, aiming at discovering the causal relations among all the variables, learns a DAG from observational data. The existing global causal discovery methods can be categorized into two types: constraint-based and score-based methods. The constraint-based methods learn the DAG by performing independent tests between variables. Popular constraint-based algorithms include PC [24], FCI [25, 26] and IC [15]. The score-based methods first define a DAG score function and then search the DAG space for a DAG with the optimal score. The score-based methods diff mainly in the search procedures, including the hill-climbing algorithm [27, 16] using greedy hill-climbing search, FGES algorithm [28] using forward-backward search, and gobnilp algorithm [29] using integer programming.

The global causal discovery methods, although asymptotically accurate, are computationally expensive. Hence, they cannot efficiently scale up to large models. To tackle this issue, various local causal discovery methods have been introduced. They typically follow a local to global approach, whereby a local causal network is first constructed for each variable and local networks are combined to form the global causal network. Growth and Shrink (GS) algorithm [17] first combines the MB of each variable to build a skeleton of global structure. It then applies the Meek rules to orient the link directions within the global structure. The max-min hill-climbing (MMHC) algorithm [16] finds MB of each variable and uses them as constraints to reduce the search space for the hill-climbing method. Based on our work on CMB discovery, we propose a local to global causal discovery approach using CMB discovery.

Our local to global causal network learning using CMB is summarized in Algorithm 4. It has three major steps. In Step 1, it uses CMB to identify the cause and effect variables w.r.t each variable (Line 3-5 in Algorithm 4).

In Step 2, the algorithm first connects each variable with it's cause and effect variables through directed links to form local causal structures. Please note as the CMB cannot ascertain the causal identity of every node, some of the links in the local causal structure may be undirected links. The algorithm then concatenates the local structures to form a global structure (Line 6 in Algorithm 4). As shown in Figure 4a, variables $\mathcal{B}$, $\mathcal{D}$, $\mathcal{E}$ and $\mathcal{F}$ are the effect variables of $\mathcal{A}$ and they are connected to $\mathcal{A}$ via $\mathcal{A} \rightarrow \mathcal{B}$, $\mathcal{A} \rightarrow \mathcal{D}$, $\mathcal{A} \rightarrow \mathcal{E}$, and $\mathcal{A} \rightarrow \mathcal{F}$. Variable $\mathcal{K}$ is the cause variable of $\mathcal{B}$ while $\mathcal{A}$, $\mathcal{C}$ are the effect variables of $\mathcal{B}$. Hence, they are connected to $\mathcal{B}$ via $\mathcal{K} \rightarrow \mathcal{B}$, $\mathcal{B} \rightarrow \mathcal{A}$ and $\mathcal{B} \rightarrow \mathcal{C}$. The local structures w.r.t $\mathcal{A}$ and $\mathcal{B}$ are concatenated together to form a global structure in Figure 4b.

Algorithm 4 Local-to-global Global Network Learning Algorithm

1: **Input:** Data: $\mathcal{D}$.
2: **Output:** The global causal structure: $\mathcal{G}$.
{Step 1: Find all the local networks}
3: **for** each $X \in \boldsymbol{V}$ **do**
4: $ID_X = \text{CMB}(\mathcal{D}, X)$;
5: **end for**
{Step2: Build global network}
6: $\mathcal{G} \leftarrow \text{BuildGlobalNetwork}(ID_X), \forall X$;
{Step3: Resolve conflicts}
7: **for** each link in $\mathcal{G}$ **do**
8: $\mathcal{G}^* \leftarrow \text{ResolveConflicts}(\mathcal{G}, ID_X), \forall X$;
9: **end for**
10: **Return:** $\mathcal{G}^*$

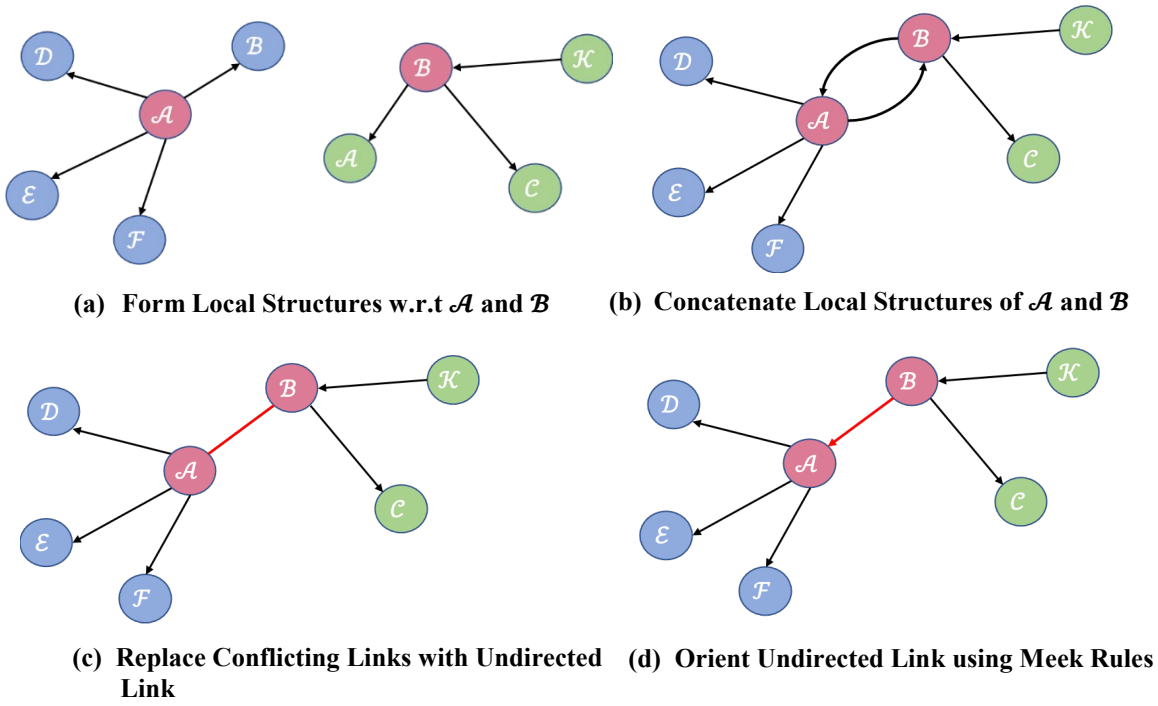

Figure 4. Illustration of the Proposed Local to Global Causal Network Learning

In Step 3, the algorithm resolves the conflicting links and orients the undirected links in the global structure from Step 2. It first replaces the conflicting directed links with undirected links. It then reorients the undirected links, following Meek rules (Line 7-9 in Algorithm 4). The Meek rules are widely used in causal discovery to orient link directions between variables. There are three fundamental Meek rules:

- **Rule 1:** For every $X, Y, Z \in \boldsymbol{V}$ such that $X \rightarrow Y$ - Z, orient $X \rightarrow Y \rightarrow Z$.
- **Rule 2:** For every $X, Y \in \boldsymbol{V}$ such that X - Y , if there exists a directed path from X to ,

orient $X \rightarrow Y$.

- **Rule 3:** For every $X, Y \in \boldsymbol{V}$ such that X - Y , if there exists non-adjacent pair $(Z, W) \in \boldsymbol{V}$ such that X - $Z \rightarrow Y$ and X - $W \rightarrow$, orient $X \rightarrow Y$.

For the undirected links that the Meek rules fail to orient, they will be left as the bi-directed links, implying the presence of latent confounding variables.

As shown in Figure 4c, $\mathcal{A} \rightarrow \mathcal{B}$ conflicts with $\mathcal{B} \rightarrow \mathcal{A}$. The algorithm replaces the links between $\mathcal{A}$ and $\mathcal{B}$ with an undirected link. In Figure 4d, since there exists $\mathcal{K} \rightarrow \mathcal{B}$ - $\mathcal{A}$, the link direction between $\mathcal{A}$ and $\mathcal{B}$ is oriented as $\mathcal{B} \rightarrow \mathcal{A}$ according to Rule 1 of the Meek rules.

3.4.3 MB Learning for Structured Classification. The goal of classification is to take an input vector $\boldsymbol{X}$ and assign a class label to the output y. Traditional classification algorithm, such as logistic regression, SVM, and neural network, learn a mapping function between the inputs and outputs. While capturing the statistical correlations be- tween inputs and outputs, the mapping function often fails to capture the structural dependencies among the inputs and outputs, as well as their uncertainties. Structured classification with PGMs addresses these limitations. PGMs such as Bayesian Networks (BNs) encode the probabilistic dependencies among the random variables with a DAG and classification/regression is performed via MAP inference over the DAG.

Specifically, structured classifiers start with constructing a DAG from training data, which includes learning both it's structure $\mathcal{G}$ and parameters $\boldsymbol{\theta}$ to capture the joint distribution $p(y, \boldsymbol{X}|\mathcal{G}, \boldsymbol{\theta})$ between feature variables $\boldsymbol{X}$ and label variable y. Structure $\mathcal{G}$ can be manually specified or learned from data. The widely-used manually specified BN structures include the Naive Bayes classifier and Tree-augmented Naive Bayes classifier as shown in Figures 5 and 6.

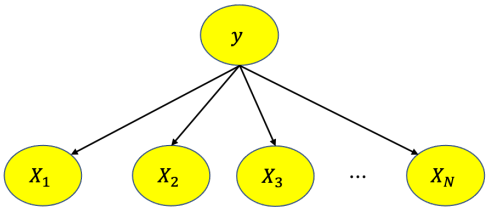

Figure 5. Example Structure of Naive Bayes Classifier

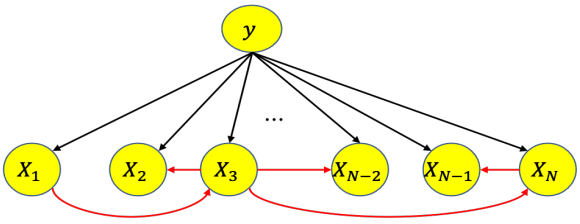

Figure 6. Example Structure of TAN Classifier

Both Naive Bayes classifier and Tree-augmented Naive Bayes classifier have strong assumptions about the BN structure. Specifically, the Naive Bayes classifier assumes the label variable y is the parent of all the feature variables $X = X_1, X_2, \cdots, X_N$ and there are no connections among the feature variables. Given $y, X_1, X_2, \cdots, X_N$ are all independent with each other. The TAN classifier relaxes the feature independence assumption by allowing direct connections among the feature variables. The dependencies between feature variables are learned from the data. As a

result, for TAN, given the label variable y, feature variables are not all independent of each other. Despite their strong assumptions, both Naive and TAN classifiers have been proven quite effective for some data.

However, for data where variables are highly dependent on each other, neither Naive and TAN can perform well. In this case, we need to learn the DAG structure from data. For this, we follow the local to global method discussed in previous section to learn causal network for structured classification.

Given the learned model, MAP inference can be performed to estimate the most likely class of label y, i.e.,

$$y^* = \arg\max_y p(y|\boldsymbol{X}, \mathcal{G}, \boldsymbol{\theta})$$

For evaluation, we compare the classification performance of two structured classifiers, Naive Bayes classifier and the causal network classifier, against the Multilayer Perceptron (MLP) classifier on the benchmark datasets from the UCI repository[4]. The empirical results are summarized Table 8, which shows that the causal network classifier outperforms Naive Bayes classifier on Nursery, Chess and Mushroom datasets but obtains worse accuracy on CMC and Adult datasets. Both structured classifiers, however, fail to outperform the MLP classifier on most of the datasets.

Table 8. Structured Classification Evaluation on UCI Datasets

Datasets	Causal Network Classifier	Naive Bayes Classifier	MLP Classifier
Nursery	0.6129	0.5969	**0.8690**
Chess	0.9016	0.7942	**0.9777**
Mushroom	**0.9823**	0.9443	0.9777
CMC	0.4198	0.4505	**0.5127**
Adult	0.6131	**0.7849**	0.7473

Our further investigation on the inadequate performance with the causal network classifier reveals that the learned DAG is inaccurate due to insufficient and imbalanced data. The inaccurate DAG, in turn, yields low classification performance. After reviewing the existing approaches and techniques for learning under insufficient or imbalanced data, we proposed a Bayesian structured classifier. Instead of learning one DAG, the Bayesian structured classifier obtains an ensemble of DAGs and combines their predictions to reduce the influence of the biased data.

Specifically, as shown in Eq. 12, different from the point-based learning of the most likely DAG, Bayesian structured classifier first constructs the posterior distribution of the DAG $\mathcal{G}$, given the training data $\mathcal{D}$, i.e., $p(\mathcal{G}|\mathcal{D})$. It then samples $p(\mathcal{G}|\mathcal{D})$ to generate multiple $\mathcal{G}_n, n = 1,2,\cdots,N$. Classification can then be performed using each $\mathcal{G}_n$ and their predictions are then combined to generate the final prediction. We employ the structure MCMC method [30] to both construct $p(\mathcal{G}|\mathcal{D})$ and sample DAGs from the posterior distribution.

[4] http://archive.ics.uci.edu/ml/datasets.php

$$
\begin{aligned}
p(y|\boldsymbol{X},\mathcal{D}) &= \int_{\mathcal{G}} p(y|\boldsymbol{X},\mathcal{G})p(\mathcal{G}|\mathcal{D})d\mathcal{G} \\
&= \mathbb{E}_{p(\mathcal{G}|\mathcal{D})}[p(y|\boldsymbol{X},\mathcal{G})] \\
&\approx \tfrac{1}{N}\textstyle\sum_{n=1}^{N} p(y|\boldsymbol{X},\mathcal{G}_n) \text{ where } \mathcal{G}_n \sim p(\mathcal{G}|\mathcal{D}) \\
y^*_{Bay} &= \arg\max_y p(y|\boldsymbol{X},\mathcal{D}) \\
&= \arg\max_y \frac{1}{N}\sum_{n=1}^{N} p(y|\boldsymbol{X},\mathcal{G}_n)
\end{aligned}
\tag{12}
$$

Equation 12. Bayesian Structured Classification

We evaluate Bayesian structured classifier on several D3M datasets. The results in Table 16 show that the Bayesian structured classifier can improve the classification performance on datasets with insufficient or imbalanced data. The improvement, however, is not applicable to every dataset.

3.4.4 MB Learning for Structured Data Imputation. Data imputation is a machine learning technique to identify and fill in the missing values in the given data before performing prediction tasks. A popular approach for data imputation is to calculate the sufficient statistics for that variable and replace all the missing values with their sufficient statistics. The sufficient statistics are usually mean or the most frequent values. However, the traditional statistical imputer only considers the statistics for the variables with missing values. It fails to account for the dependencies between the variables with missing values and other variables.

Hence, we propose to construct a structured imputer. The structured imputer learns a BN model from incomplete data and infers the missing values via MAP inference. With the incomplete data, conventional MLE estimation of a BN structure cannot apply. We propose to employ the hard Structure Expectation-Maximization (SEM) method to learn a DAG to capture the relationships among different variables. The hard SEM includes two steps. In the E-step, it infers the most likely values of the variables with missing values, given the current DAG model and values of other variables, and then completes the data using the inferred values. In the M-step, given the completed data, it uses the standard MLE method to learn the DAG structure by maximizing it's log-likelihood. The E step and M step repeat alternately until their convergence. Given the learned DAG, a MAP inference can be performed to infer the missing values, given the observed values.

We compare our structured imputer with the statistic imputer on multiple D3M datasets. For this, we first employ the two imputers to complete the data. We then employ the random forest algorithm to perform classification on the imputed data. The imputers are evaluated in terms of both their contribution to the classification performance and their computational efficiency. The

empirical results are summarized in Table 9, where the missing value ratio shows the percentage of the missing values in the data. Table 9 shows that compared to the statistic imputer, our structured imputer can only marginally improve the classification performance. It's computational complexity, on the other hand, is much higher due to the iterative learning of the BN model.

Table 9. Structured Imputation Evaluation on D3M Datasets

Datasets	Missing Value Ratio	Structured Imputer	Statistic Imputer
38	2.19%	**0.9345**	0.9248
57	2.27%	**0.9576**	0.9517
185	0.08%	**0.6438**	0.6433

4 RESULTS AND DISCUSSION

In this section, we first briefly discuss the integration of the methods we developed for local structure learning into D3M environment. We then discuss the results of performance evaluation of our methods for different machine learning tasks on D3M datasets.

4.1 Integration into D3M Environment

To meet D3M's software and evaluation requirements, we develop and implement 3 feature selection primitives, 2 classification primitives, and 1 regression primitive and integrate them into the D3M environment. The 3 feature selection primitives were initially implemented in Matlab and were later re-implemented in Python to resolve the conflicts with packages used by some TA2 teams. We also remove the semantic schema dependency in our primitives to meet the latest D3M environment requirements. All our primitives were implemented following the D3M scheme and were integrated into the final D3M environment. For each primitive, we construct one or more pipelines to show how to use our primitives. All the pipelines passed the evaluation test and were successfully integrated into the final D3M environment. The packages[5,6] for our primitives are easy to install through the pip command, with no dependency on other primitives in the D3M environment.

4.2 Performance Evaluation

We then conduct a performance evaluation of our primitives and pipelines within D3M environment on D3M datasets against D3M designated baselines and primitives from other team members.

[5] https://gitlab.com/N.Yin/rpi-d3m-part2
[6] https://github.com/naiyuyin/rpi-d3m-primitives

4.2.1 Structured Feature Selection. To evaluate the performance of structured feature selection on D3M datasets, we implemented a STMB based structured feature selection primitive, within which we include four methods to perform the independence tests, including the three Bayesian independence tests discussed in Section 3.2 and a standard mutual information based method. The three Bayesian independence tests include 1) pseudo Bayesian MI-based independence test; 2) empirical Bayesian MI-based independence test; and 3) Bayes Factor independence test. A hyper-parameter is included in the primitive to allow users to select an independence test method. Besides STMB, we also provide two more feature selection primitives: 1) score-based simultaneous markov blanket (S2TMB) [31], which performs score-based instead of constraint-based MB learning; 2) joint mutual information method [32], which is a standard feature selection technique.

We apply these feature selection primitives to classification and regression tasks, and compare their performance against D3M designated baselines and the best performance achieved by other teams on the latest leaderboard[7]. Performance metric is specified for each D3M dataset: for classification tasks, F1-score is employed; for regression tasks, mean squared error (MSE) is employed[8]. Pipelines are constructed within the D3M environment and hyper-parameters of pipelines are tuned based on training set. We also include the feature compression ratio as an efficiency metric. Results are summarized in Tables 10 and 11, where performance of our pipelines are bolded if they are ranked top 1 on the leaderboard.

In total, our pipelines outperform rank 1 performance on 6 out of 12 datasets. In the meantime, our pipelines outperform the baseline performance on majority of the datasets. Evaluation results show that our structured feature selection primitives can achieve competitive performance with only a small set of features employed. For example, on dataset 185, STMB primitive outperforms the rank 1 performance (F1-score 77.2%) with only 13 out of 19 features selected. S2TMB primitive doesn't have results on dataset 26 because no feature is selected.

To further demonstrate the effectiveness of our Bayesian independence tests, we compare the performance of STMB with Bayesian independence tests (Bayesian STMB) against STMB with standard MI-based independence test (Standard STMB). We follow the same experimental settings and employ the same evaluation metrics as we introduced above. Results are shown in Table 12. On both dataset 26 and dataset 38, Bayesian STMB primitive outperforms the standard STMB in both accuracy and feature compression efficiency. For example, on dataset 26, Bayesian STMB primitive with both pseudo Bayesian MI and Bayes Factor achieves rank 1 performance (MSE $3.6e-5$) with only 1 out of 28 features selected, while standard STMB primitive achieves MSE $76.7e-5$ with 25 features selected. Bayesian STMB with empirical Bayesian MI doesn't have results on dataset 26 because no feature is selected.

[7] The latest leaderboard on D3M datasets is for the dry run on December, 2020: https://leaderboarddecember2020dryrun.datadrivendiscovery.org/

[8] Root Mean Squared Error (rMSE) was actually employed as the performance metric on dataset 26 during the evaluation.

Table 10. Feature Selection Evaluation for Classification

Datasets	Task	Ours			Leaderborad	
		Methods	Compression Ratio	F1-score	Baseline	Best
1100	Classification	STMB	11/8	0.494	0.414	0.484
		S2TMB	11/4	0.362		
		JMI	11/8	**0.513**		
313	Classification	STMB	103/91	0.446	0.191	0.522
		S2TMB	103/3	0.397		
		JMI	103/16	**0.574**		
185	Classification	STMB	19/13	**0.772**	0.691	0.727
		S2TMB	19/6	0.704		
		JMI	19/13	0.755		
1491	Classification	STMB	64/62	0.828	0.694	0.918
		S2TMB	64/64	0.788		
		JMI	64/3	0.658		
Acled	Classification	STMB	29/11	0.730	0.848	0.921
		S2TMB	29/12	0.767		
		JMI	29/12	0.810		
27	Classification	STMB	14/7	0.277	0.319	0.320
		S2TMB	14/11	0.264		
		JMI	14/5	0.269		
57	Classification	STMB	29/11	**1.000**	0.843	0.991
		S2TMB	29/10	0.987		
		JMI	29/19	0.964		
186	Classification	STMB	10/6	0.227	0.192	0.363
		S2TMB	10/5	0.161		
		JMI	10/1	0.191		

Table 11. Feature Selection Evaluation for Regression

Datasets	Task	Ours			Leaderborad	
		Methods	Compression Ratio	MSE	Baseline	Best
26	Regression	STMB	28/1	**3.6e-5**	0.585	$12.1e-5$
		S2TMB	—	—		
		JMI	28/1	$12.1e-5$		
196	Regression	STMB	7/6	6.674	7.371	5.354
		S2TMB	7/3	5.171		
		JMI	7/5	**4.713**		
534	Regression	STMB	11/3	24.211	19.742	15.905
		S2TMB	11/6	22.759		
		JMI	11/9	24.868		
207	Regression	STMB	15/3	$3.0628e6$	$6.9857e6$	31351.6
		S2TMB	15/4	$3.3273e6$		
		JMI	15/6	$3.0870e6$		

Due to the substantial performance improvement, our feature selection primitives were widely adopted by TA2 teams. Specifically, for each primitive, we report the number of TA2 teams that used it, the number of D3M datasets that it is evaluated on, and the number of submitted pipelines that include the primitive. In addition, we measure the performance of these submitted pipelines in terms of the number of pipelines that beat the baseline performance on the leaderboard. These statistics are summarized in Table 13. As we can see, our feature selection primitives are widely used by TA2 teams. For example, the STMB feature selection primitive is used by 3 TA2 teams, is incorporated into 269 pipelines, and outperforms the baseline 99 times.

Table 12. Bayesian STMB versus Standard STMB on D3M Datasets

Datasets	Task	Ours			Leaderboard	
		Independence Test	Compression Ratio	F1-score/MSE	Baseline	Best
57	Classification	Standard MI	29/11	**1**	0.8430	0.9910
		Pseudo Bayesian MI	29/25	**1**		
		Empirical Bayesian MI	29/25	**1**		
		Bayes Factor	29/25	**1**		
38	Classification	Standard MI	29/27	0.9333	0.8500	0.9450
		Pseudo Bayesian MI	29/14	**0.9565**		
		Empirical Bayesian MI	29/27	0.9333		
		Bayes Factor	29/11	0.9556		
26	Regression	Standard MI	28/25	$76.7e-5$	0.585	12.1e-5
		Pseudo Bayesian MI	28/1	**3.6e−5**		
		Empirical Bayesian MI	—	—		
		Bayes Factor	28/1	**3.6e−5**		

Table 13. TA2 Usage on Feature Selection Primitives

Primitives	Usage Statistic			Performance
	#TA2	#Dataset	#Pipeline	#Pipeline beating baseline
STMB	3	64	269	99
S2TMB	2	45	181	78
JMI	2	47	124	52

4.2.2 Structured Classification and Regression. For structured classification and regression, we implement a causal network classifier and a causal network regressor. The causal network classifier was implemented based on the PC algorithm [15] from py-causal package[9]. The causal network regressor was implemented based on the FGES algorithm [33] from the same package. Our structured classifiers and regressor are evaluated on several D3M datasets. For classification, we report F1-score as the performance metric. For regression, we report mean squared error (MSE). Hyper-parameters of pipelines are optimized based on training set. For comparison, the baseline performance and the best performance achieved by other teams on the latest D3M leaderboard are also reported. The performance of the causal network classifier and the causal network regressor are shown respectively in Tables 14 and 15.

[9] https://github.com/bd2kccd/py-causal

Table 14. Structured Classification Evaluation

Datasets	Task	Causal Network Classifier F1-score	Leaderboard	
			Baseline	Best
57	Classification	0.549	0.843	0.991
1100	Classification	0.228	0.414	0.484
27	Classification	0.118	0.319	0.320
185	Classification	0.487	0.691	0.727
186	Classification	0.236	0.192	0.373
4550	Classification	**1**	0.986	1

Table 15. Structured Regression Evaluation

Datasets	Task	Causal Network Regressor MSE	Leaderboard	
			Baseline	Best
207	Regression	$6.0415e6$	$6.9857e6$	31351.6
196	Regression	13.690	7.371	5.354
26	Regression	0.148	0.585	$12.1e-5$
534	Regression	23.310	19.742	15.905

Empirical results in Tables 14 and 15 show that the causal network classifier achieves the best performance on 4550 dataset and outperforms baseline performance on dataset 186. The causal network regressor outperforms the baseline on 207 and 26 datasets. But overall, our causal network classifier and regressor are not competitive. They fail to achieve comparable performance even to the baseline on datasets 57, 1100, 27, and 185. Hence, only one TA2 system uses our causal network classifier. TA2 usage of causal network regressor is unknown as it is submitted at the final phrase of the project.

We mention previously that the inadequate performance with our structured classifier is mainly due to insufficient or imbalanced data, which leads to inaccurate DAG and hence low performance. For example, 1100 dataset has three classes. Training data distribution of the three classes are 51.59%, 29.63% and 18.78%. It is clear training data for class 3 is significantly insufficient. Hence, the learned model tends to mis-classify class 3 samples as class 1. To prevent such erroneous prediction, we implement the proposed Bayesian structured classifier following Eq. 12 , whereby we construct an ensemble of structured classifiers to jointly perform classification. Table 16 shows the comparison of standard structured classifier against the Bayesian structured classifier on three D3M datasets 1100, 27, and 186.

Table 16. Bayesian Structured Classifier versus Standard Structured Classifier

Datasets	Task	Structured classifier		Leaderboard	
		Standard	Bayesian	Baseline	Best
1100	Classification	0.228	0.421	0.414	0.484
27	Classification	0.118	0.118	0.319	0.320
186	Classification	0.236	0.148	0.192	0.373

Empirical results in Table 16 show that the Bayesian structured classifier can substantially improve the classification performance on the 1100 dataset; the Bayesian structured classifier increases the classification accuracy by 19.3% and outperforms the baseline on the leaderboard. The Bayesian structured classifier, however, cannot achieve good performance improvement on severely imbalanced datasets such as 27 and 186.

5 CONCLUSIONS

Through this research, we developed several methods for learning local structure (Markov Blanket) for a target variable. We then apply the MB learning methods to structured feature selection, local and global causal network learning, structured classification, and structured data imputation. We evaluated their performance on benchmark datasets and they demonstrated competitive performance against state of the art methods. In addition, we implemented our methods as machine learning primitives, following D3M scheme, and integrated them into the latest D3M environment. Evaluations within the D3M environment on D3M datasets demonstrated the competitive performance of our primitives against the baselines and the best methods from other teams. Particularly, our structured feature selection primitives significantly outperformed the baseline, and achieved rank 1 performance on many D3M datasets. As a result, these primitives were adopted by several TA2 teams, and were incorporated into a large number of pipelines. Our structured classification and regression primitives also achieved good performance on a few D3M datasets.

6. REFERENCES

[1] J. M. Pena, R. Nilsson, J. Bjorkegren, and J. Tegner, "Towards scalable and data efficient learning of markov boundaries," *International Journal of Approximate Reasoning*, vol. 45, no. 2, pp. 211–232, 2007.

[2] S. Fu and M. C. Desmarais, "Fast markov blanket discovery algorithm via local learning within single pass," in *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 96–107, Springer, 2008.

[3] T. Gao, Z. Wang, and Q. Ji, "Structured feature selection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4256–4264, 2015.

[4] J. H. McDonald, Handbook of biological statistics, vol. 2. sparky house publishing Baltimore, MD, 2009.

[5] S. Mukherjee and T. P. Speed, "Markov chain monte carlo for structural inference with prior information," *University of California; Berkeley*, 2007.

[6] K. Korb and A. Nicholson, *Bayesian Artificial Intelligence*. Chapman and Hall, 2nd ed., 2010.

[7] M. Hutter, "Distribution of mutual information," in *Advances in neural information processing systems*, pp. 399–406, 2002.

[8] T. Minka, "Estimating a dirichlet distribution," 2000.

[9] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the american statistical association*, vol. 90, no. 430, pp. 773–795, 1995.

[10] T. Gao and Q. Ji, "Local causal discovery of direct causes and effects," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2512–2520, 2015.

[11] G. F. Cooper, "A simple constraint-based algorithm for efficiently mining observational databases for causal relationships," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 203–224, 1997.

[12] S. Mani and G. F. Cooper, "A study in causal discovery from population-based infant birth and death records.," in *Proceedings of the AMIA Symposium*, p. 315, American Medical Informatics Association, 1999.

[13] C. Silverstein, S. Brin, R. Motwani, and J. Ullman, "Scalable techniques for mining causal structures," *Data Mining and Knowledge Discovery*, vol. 4, no. 2, pp. 163–192, 2000.

[14] S. Mani and G. F. Cooper, "Causal discovery using a bayesian local causal discovery algorithm," in *MEDINFO 2004*, pp. 731–735, IOS Press, 2004.

[15] J. Pearl, "Causality: models, reasoning, and inference," *Econometric Theory*, vol. 19, no. 46, pp. 675–685, 2003.

[16] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine learning*, vol. 65, no. 1, pp. 31–78, 2006.

[17] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, p. 505–511, 1999.

[18] J.-P. Pellet and A. Elisseeff, "Using markov blankets for causal structure learning.," *Journal of Machine Learning Research*, vol. 9, no. 7, 2008.

[19] Y. Zhang, J. Wu, and J. Cai, "Compact representation for image classification: To choose or to compress?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 907–914, 2014.

[20] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3025–3032, 2013.

[21] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2007 results," 2007.

[22] A. Vedaldi and A. Zisserman, "Sparse kernel approximations for efficient classification and detection," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2320– 2327, IEEE, 2012.

[23] J. Sa´nchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *CVPR 2011*, pp. 1665–1672, IEEE, 2011.

[24] P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richard- son, *Causation, prediction, and search*. MIT press, 2000.

[25] P. Spirtes, C. Meek, and T. Richardson, "Causal inference in the presence of latent variables and selection bias," in *UAI*, 1995.

[26] J. Zhang, "On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias," *Artificial Intelligence*, vol. 172, no. 16-17, pp. 1873– 1896, 2008.

[27] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[28] D. M. Chickering, "Optimal structure identification with greedy search," *Journal of Machine Learning Research*, 2002.

[29] T. Jaakkola, D. Sontag, A. Globerson, and M. Meila, "Learning bayesian network structure using LP relaxations," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 358–365, PMLR, 2010.

[30] D. Madigan, J. York, and D. Allard, "Bayesian graphical models for discrete data," *International Statistical Review/Revue International de Statistique*, pp. 215–232, 1995.

[31] T. Gao and Q. Ji, "Efficient score-based markov blanket discovery," *International Journal of Approximate Reasoning*, vol. 80, pp. 277–293, 2017.

[32] H. Yang and J. Moody, "Feature selection based on joint mutual information," in *Proceedings of international ICSC symposium on advances in intelligent data analysis*, vol. 23, Citeseer, 1999.

[33] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour, "A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images," *International journal of data science and analytics*, vol. 3, no. 2, pp. 121–129, 2017.

Appendix - Publications and presentations

List of Presentations

- Local Structure Learning and Knowledge Augmented Learning-D3M kick-off meeting
  Qiang Ji (DARPA D3M Kickoff Meeting, 03/28/2017)
- Local Structure Learning for Feature Selection
  Qiang Ji (DARPA D3M Winter Evaluation, 01/29/2018)
- Local Structure Learning for Feature Selection
  Qiang Ji (DARPA D3M Site Visit, 04/04/2018)
- Local Structure Learning for Feature Selection
  Qiang Ji (DARPA D3M Summer Evaluation, 06/11/2018)
- Project Progress Report
  Qiang Ji (DARPA Tele Meeting, 05/17/2019)
- Team Brief: Rensselaer Polytechnic Institute
  Qiang Ji (TA1 Summer 2019 Workshop, 06/10/2019)
- Team Brief: Rensselaer Polytechnic Institute
  Qiang Ji (TA1 Winter 2020 Workshop, 01/17/2020)
- Team Brief: Rensselaer Polytechnic Institute
  Qiang Ji (TA1 Summer 2020 Workshop, 07/2020)
- TA1 Primitives Status and Cleanup: Rensselaer Polytechnic Institute
  Qiang Ji (TA1 Online Meeting, 03/27/2021)

List of Publications

- Label Error Correction and Generation Through Label Relationships
  Zijun Cui, Yong Zhang, and Qiang Ji (AAAI, 2020)
- Knowledge Augmented Deep Neural Networks for Joint Facial Expression and Action Unit Recognition
  Zijun Cui, Tengfei Song, Yuru Wang, and Qiang Ji (NeurIPS 2020)
- Bayesian Approaches for Robust Constraint-based Causal Discovery under Insufficient Data, *Zijun Cui, Naiyu Yin, Yuru Wang, and Qiang Ji* (Under review by NeurIPS 2021)
- DAGs with No Curl: An Efficient DAG Structure Learning Approach
  Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji (ICML, 2021)
- Efficient Non-parametric DAG Structure Learning
  Naiyu Yin, Tian Gao, Yue Yu, and Qiang Ji (Under review by NeurIPS 2021)

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

$\mathcal{G}$	Structure of a Bayesian Network
$\boldsymbol{\theta}$	Probability distribution parameters
$X \perp\!\!\!\perp Y$	Variable X is independent of variable Y
$X \not\perp\!\!\!\perp Y$	Variable X is dependent of variable Y
$\boldsymbol{MB}_T$	Markov Blanket of node T
$\boldsymbol{PC}_T$	Parents and children (PC) set of node T
D	Observational Data
BN	Bayesian Network
DAG	Directed Acyclic Graph
MB	Markov Blanket
STMB	Simultaneous Markov Blanket
CMB	Causal Markov Blanket
ID_T	Causal identities of nodes w.r.t node T
id_T	Individual causal identity of nodes w.r.t node T
MLE	Maximum Likelihood Estimation
MI	Mutual Information
MSE	Mean Squared Error
rMSE	root Mean Squared Error
mAP	mean Average Precision