Noname manuscript No.
(will be inserted by the editor)

Local-to-global causal discovery using minimum message length

First Author · Second Author · Third Author

Received: date / Accepted: date

Given that we can learn Markov blanekts using any one of the three proposed MML methods (i.e., $MBMML+CPT, MBMML+NB, MBMML+RANDOM$), the next step is to learn the local structures within the learned Markov blankets. There are several ways of doing so:

(a) Give the learned Markov blankets to CaMML with no additional information about the local structures. CaMML will output n local DAGs, one for each node. In the end, somehow resolve local DAGs conflicts so that they can be unified into a global DAG.
(b) Approximate certainties of directed/indirected edges in Markov blankets using bootstarpping [Friedman *et al.*, 1999]. Give CaMML the learned Markov blankets and the estimated certainties as arc prior. Obtain n local DAGs and do as above.
(c) For each learned Markov blanket, calculate MML score of all possible Markov blanket polytrees (MBPs). Choose the one with the shortest message length as the most probably MBP. Repeat the process to obtain n MBPs. In the end, do as above.
(d) Use Metropolis-Hasting algorithm to approximate the posterior distribution of the space of MBPs within each learned Markov blanket. Alternatively, calculate the exact posterior distribution if there are not many MBPs. Repeat this process for each learned Markov blanket. At the end, we have n dependent posterior distributions. Then we have an optimization problem. The task is to find n non-conflicting MBPs such that the sum of their posterior probability is maximum (or MML score is minimum).

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

1 Related works

[?] derived an efficient formula for computing the joint distribution of a BN structure B_s and a given data set D

$$p(B_s, D) = p(B_s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!. \tag{1}$$

The probability $p(B_s)$ can be considered as a constant if the structure prior is assumed to be uniform. The formula was developed based on the assumptions of (1) no missing values, (2) i.i.d. samples, (3) discrete random variables and (4) uniform parameter values. The authors also calculated the computational complexity of the above equation. This formula was then revised to incorporate with the K2 algorithm presented with an additional assumption that a node ordering is available to K2.

[Friedman and Koller, 2000] presented mcmc sampling the total ordering space instead of the dag space, which is much larger and peaky. Since a BN is just a collection of parents set for all nodes, and each total ordering is consist with several dags, the probability $P(D \mid ordering)$ is the sum of probabilities over all dags consistent with the ordering, which is equivalent to the sum of scores for each node over its all possible parents set consistent with the ordering. The authors stated there is an efficient way of calculating the score. The authors assumed a uniform prior over orderings, so dags that are consistent with more orderings are more likely. The operations stated for mcmc to move through the sampling space are cut deck and switch two nodes with some probabilities. For cut deck, each possible cut is equal likely. The motivation of this paper is not to learn a single best dag, but to obtain a better estimation of features such as edge or markov blanket.

[Gillispie and Perlman, 2001] presented a way of enumerating all labelled essential graphs.

[Riggelsen, 2005] used Markov blanket MCMC to learn Bayesian networks. Its difference from previous MCMC works on structure learning is that MB-MCMC focuses on sampling edges within MBs to avoid being stuck in locally in the enormous large DAG space.

[Nägele *et al.*, 2007] presented a local-to-global BN learning algorithm similar as MMHC. It starts by putting up an undirected skeleton over all variables by using MMPC. Then it focuses on learning local structures over a target, its distance 1 and 2 neighbours. The learned local structure is then trimmed to the target's MB and MB DAG only. This process is repeated to all variables so the output is a collection of partially overlapping local MB DAGs. To resolve edge conflicts, the authors estimated direction and edge existence based on bootstrapping.

[Pellet and Elisseeff, 2008] assumed MBs are correctly learned, used moral graph to help identifying spouses and hence learn a causal model using constraint-based methods.

[Riggelsen, 2008] proposed a new score (but similar as BDe) for learning Bayesian networks. The only difference is that this new score finds the optimal pair of DAG and parameters for a given dataset, rather than just finding the optimal DAG as BDe does (according to the author). The parameters are estimated using MAP estimation, and hence this new score is called MAP BN.

[Pensar *et al.*, 2014] learned Markov networks (undirected) without assuming chordalysis, proposed a new scoring function, used Markov blanket concept in undirected graphs (which are just neighbours of a node), talked about global optimization of a finding a graph consistent with the B_X.

[Gao *et al.*, 2017] presented a local-to-global bn learning algorithm. The algorithm is based on [Gao and Ji, 2017]'s Markov blanket discovery algorithm to find the local structure around an arbitrary target, then expand the structure gradually to obtain a global bn.

2 Merging Markov blankets into a global DAG

The first objective is to find a DAG such that the sum of the edit distance between the learned MBs and the read-off MBs from this DAG is minimum. We started by creating an empty DAG then for each node, its learned MB is connected to it as direct neighbours without assigning directions to the edges. The resulting graph is a moral graph of the true DAG if the learned MBs are correct. By the next proposition, we do not need to apply symmetry enforcement on the learned MBs.

Proposition 1 *Let* $\{MB_i\}$ *and* $\{MB_i^s\}$ *be the set of the learned MBs for all nodes without and with symmetry enforcement on the results respectively.* MB_i *nodes are added as the direct neighbours of* X_i *to produce an undirected graph* G*. Then* $MB_i^G = MB_i^s$*.*

The proof is trivial. This way we obtain a DAG that has a low total MB edit distance to the learned MBs. To imporve this initial DAG, we must first assign directions to existing arcs since our scoring metric (MML) prefers to know the parents set of each node.

Proposition 2 *Every undirected graph can be made into a DAG.*

The above proposition is trivial. Take an undirected graph and pick a random node, lift it up then we will have node ordering. Direct every edge from top to down, then we will end up with a DAG. This proposition ensures the current undirected graph G can be made into a DAG.

We can do the following:

- Identify the set C of three nodes cliques in G.
- Assign high confidence to edges between each node and its first found MB node, and store in E^*. Noticing from experimental results, we are confidence that the first found MB node is in the neighbour of the target. Experimental results have shown that the certainty of the first found node being a true positive varies from $(0.75, 0.85)$ for models with 20 nodes, maximum 4 arity for each node and maximum number of parents $\{2, 3, 4, 5, 6\}$ given 100 samples.
- List all arcs appeared in the MBPTs and count the number of times they occur, then assign directions to edges in G according to the direction mode, if draw then leave undirected.

- Alternatively, we could feed the initial DAG G to camml. But camml could be improved both in speed and perhaps in accuracy.

Proposition 3 *This is more like a conjecture! For $n \longrightarrow \infty$, the learned DAG by adding MB_i as neighbours of X_i contains only false positives, there is no false negative.*

This could save mcmc time for not to consider adding more arcs, but focus on deleting arcs. And reduce the initial sampling space by admitting arcs in G. But I still need directions to start with.

Once this initial DAG is created, the next step is to consider edge directions especially those that form a collider. The initial DAG is denser than the true DAG because spouses (if there are any) are added as direct neighbours of each node. If we can identify variables X, Y, Z such that $ind(X, Y)$ but $dep(X, Y \mid Z)$, then the edge between X and Y should be removed. To identify this, we could use interaction information, which is defined as $I(X, Y, Z) = I(X, Y) - I(X, Y \mid Z)$. In principle, if a DAG contains only three nodes and Z is a common child of X and Y, then $I(X, Y, Z) < 0$. But there are some issues using interaction information.

The first issue is that if $I(X, Y, Z) < 0$ then any one of these three nodes can be the common child, because $I(X, Y, Z) = I(X, Y) - I(X, Y \mid Z) = I(X, Z) - I(X, Z \mid Y) = I(Z, Y) - I(Z, Y \mid X)$. Hence, interaction information does not give any information about which is more likely to be the common child. The second issue is it is possible to have $I(X, Y, Z) < 0$ when$dep(X, Y)$ and $dep(X, Y \mid Z)$, as long as the dependency between X and Y is weak.

Proposition 4 *If $I(X, Y, Z) < 0$, then either there is a weak directed arc between the two parents, or there is a weak indirect path between the two paretns, or they are independent.*

Not sure if anyone has proved this in a general DAG.

3 Improve CaMML's efficiency by MB

camml takes structure priors and based on these priors, adjust the mml score of toms. If a given prior is true with high confidence, then the toms consistent with this prior is more likely to be sampled with some probability. But does this reduce the tom space for sampling? Or does it improve mixing/convergence of MCMC? Other people who used mcmc to sample ordering considered improve convergence/mixing and reduce the burn in period. camml doesn't talk about burn in. But camml seems fixed the number of samples drawn from tom space, perhaps we could reduce the number of samples required, and make it dynamic according to something. Is it possible to avoid some regions in the tom space and group some the picky points into one group that consistent with the given mb prior so that the total sampling space is reduced and the high poterior points are group near each other.

Camml's performance is good. It beats MBCPT for MB discovery with quite a margin. But it is really slow on large models, tested on 50 variables, took about 10 mins to finish. Definitely aim for improving its efficiency using mb results. But since MB results are not particularly accurate, give its results to camml as prior will affect camml's reconstruction accuracy. Maybe it's ok to sacrifice accuracy for efficiency.

3.1 Testing CaMML's response to various prior

We have done some testing on CaMML's reponse to various prior. The purpose of these experiments is to work out the impact of the best (correct MB info with high confidence) and worst (incorrect MB info with high confidence) priors. The experimental settings are 6 random DAGs with $30-4-5-1-500$, and the reporting statistics is equivalent class edit distance with 0.95 confidence interval. The results are as the following:

- with no prior, ed = 9.5+-2.7;
- with true edge prior and confidence 1, ed = 0.33 +- 0.65 (non-overlapping);
- with true edge prior and confidence 0.95, ed = 3.5 +- 2.9;
- with true edge prior and confidence 0.9, ed = 4.5+-3.6;
- with true edge prior and confidence 0.8, ed = 5.3 +- 4;
- with true edge prior and confidence 0.7, ed = 5.7 +- 4.1;
- with true edge prior and confidence 0.5, ed = 5 +- 3.9.

It seems that as long as the given priors are correct, even with 0.5 confidence the accuracy still improved though not statistically significant. Now, we test on giving false priors. We randomly generated 20 false arcs and give them to camml as undirected arc priors with different confidence levels. The results are as the followings:

- with false priors and confidence 1, ed = 37 +- 4.9 (non-overlapping);
- with false priors and confidence 0.8, ed = 10 +- 3.4;
- with false priors and confidence 0.5, ed = 8.7 +- 2.4.

Now, we mix the correct and incorrect undirected arc priors and test the impact on different confidence levels. From previous results, we know that for models with this complexity and 500 samples, MML+CPT's MB discovery precision is about 0.9 and recall 0.56. Given that we currently can only treated all MB nodes as directly connected with the target, we mix 0.7 correct undirected arc priors with 0.3 incorrect undiretec arc priors. Given that once correct prior confidence is less than 0.9 and incorrect prior confidence is less than 0.8, camml does similarly as no priors given, we start experiments with these confidence levels for correct and incorrect priors respectively. The results are as the followings:

- 0.7 true 0.3 false with confidence 0.9 and 0.8 respectively, ed = 7.6 +- 2.8;
-

3.2 Enumerating all sets of Markov blankets

Definition 1 A *clique* is a subset of nodes in an undirected graph where every two distinct nodes are adjacent.

We use Q_m to denote an m-clique.

Definition 2 A *simplicial node* in an undirected graph is a node whose neighbours form a clique.

Definition 3 A graph F is *recursively simplicial* if it contains a simplicial node X_i and the induced subgraph $F[X \setminus \{X_i\}]$ is recursively simplicial.

Definition 4 A *simplicial node ordering* of a recursively simplicial graph F is a sequence of nodes $\{X_1, \dots, X_n\}$, where X_i is a simplicial node in the induced subgraph $F[\{X_{i+1}, \dots, X_n\}]$.

Definition 5 An *m-cycle* in an undirected graph is a sequence of nodes $\{X_1, \dots, X_{m+1}\}$ where $X_1 = X_{m+1}$ and all the other nodes are distinct.

Definition 6 A graph is chordal if each m-cycle for $m \geq 4$ has a chord.

Proposition 5 *The following properties of an undirected graph F are equivalent:*

1. *F is chordal.*
2. *F is recursively simplicial.*
3. *F can be oriented to obtain a DAG G s.t. the moral graph of G is F.*
4. *F can be oriented to obtain a DAG G s.t. F and G imply the same conditional independences.*

Corollary 1 *If F is a chordal graph, there exists a DAG G obtained by orienting edges of F s.t.* $N_X^F = B_X^G$.

Proof From Proposition 5, we can obtain a DAG G by orienting the edges of F according to a fixed simplicial node ordering. Since $X_i \perp\!\!\!\perp_P S \mid N_i^F$ and $X_i \perp\!\!\!\perp_P S \mid B_i^G$, it implies $N_i^F = B_i^G$ for all $X_i \in X$. □

Proposition 6 *Let* $\mathcal{F}|_c = \{F \mid F \text{ is chordal}\}$ *and* $\mathcal{G}^* = \{G \mid B_X^G = N_X^F\}$. *Define a function*

$$f : \mathcal{F}|_c \to \mathcal{G}^*$$

such that for a given simplicial node ordering $\{X_1 \dots X_n\}$, $X_j \to X_i$ *for all* X_i*'s neighbours in the induced subgraph* $F[X \setminus \{X_1, \dots, X_{i-1}\}]$. *Then* f *is an injective but non-surjective function.*

Proof It is obvious that $f(F_1) \neq f(F_2)$ if $F_1 \neq F_2$, so f is injective. For a DAG $G \in \mathcal{G}^*$ s.t. $B_X = \{\{X_2, X_4\}, \{X_1, X_3\}, \{X_2, X_4, X_5\}, \{X_1, X_3, X_5\}, \{X3, X_4\}\}$, there is no chordal graph F satisfies $N_X^F = B_X^G$, so f is not surjective. □

Definition 7 A graph is *weak recursively simplicial* if it contains a simplicial node X_i and the subgraph S over $X \setminus \{X_i\}$ is weak recursively simplicial, where $E(F[X \setminus \{X_i\}]) \setminus E(S) = \{X_j - X_k \mid X_j, X_k$ are not simplicial nodes in $F[X \setminus \{X_i\}]\}$.

If a graph is recursively simplicial it is also weak recursively simplicial, but not vice versa.

Definition 8 A *weak simplicial node ordering* of a weak recursively simplicial graph F is a sequence of nodes $\{X_1, \dots, X_n\}$ s.t. $X_i \prec X_j$ if and only X_i is a simplicial node in the subgraph S over $X \setminus \{X_i\}$ whilst X_j is not, where $E(F[X \setminus \{X_i\}]) \setminus E(S) = \{X_j - X_k \mid X_j, X_k$ are not simplicial nodes in $F[X \setminus \{X_i\}]\}$.

Proposition 7 *Let* $\mathcal{F}^* = \{F \mid F$ *is weak recursively simplicial*$\}$ *and* $\mathcal{G}^* = \{G \mid B_X^G = N_X^F\}$. *Define a function*

$$f : \mathcal{F}^* \to \mathcal{G}^*$$

such that for a given weak simplicial node ordering $\{X_1 \dots X_n\}$,

- $X_j \to X_i$ *for all* X_i*'s neighbours in the induced subgraph* $F[X \setminus \{X_1, \dots, X_{i-1}\}]$,
- $E(F[X \setminus \{X_i\}]) \setminus E(S) = \{X_j - X_k \mid X_j, X_k$ *are not simplicial nodes in* $F[X \setminus \{X_i\}], \forall X_j, X_k \in N_i^F\}$.

Then f *is a bijective function.*

Proof Let $F_1, F_2 \in \mathcal{F}^*$ be two graphs different by an edge $e = X_j - X_k$ where $e \in E(F_1)$ but $e \notin E(F_2)$. This implies $X_j \in N_k^{F_1}$ and $X_j \notin N_k^{F_2}$, hence $X_j \in B_k^{f(F_1)}$ and $X_j \notin B_k^{f(F_2)}$ unless there is a node X_i s.t. $X_j \to X_i \leftarrow X_k$ in $f(F_2)$. By definition of f, $X_i \prec \{X_j, X_k\}$ in a weak recursively node ordering, and there must be an edge $X_j - X_k$ for otherwise X_i will not be a collider in $f(F_2)$. But the assumption is $e \notin E(f(F_2))$, so f is injective.

To show that f is surjective, we prove by contradiction. Assuming $\exists G \in \mathcal{G}^*$ s.t. $F = f^{-1}(G) \notin \mathcal{F}^*$. It implies F does not have a weak simplicial node ordering, so $G = f(F)$ is a hybrid graph. To make G into a DAG we must introduce colliders in the undirected part, but then $B_X^G \neq N_X^F$ so there is a contradiction. □

The proof of the above proposition did not rely on how the function f is defined. Not sure if this is right!

Corollary 2 $|\mathcal{F}^*| = |\mathcal{G}|_{B_X}|$

Proof The function $f : \mathcal{F}^* \to \mathcal{G}|_{B_X}$ is defined in such a way that $N_X^F = B_X^{f(F)}$. f is bijective implies its domain and codomain have the same cardinality. □

Corollary 3 *DAGs in the same Markov equivalent class produce the same Markov blanket sets* B_X.

Proof If two DAGs G_1 and G_2 are Markov equivalent, they have the same skeleton and the same set of colliders. This implies $B_i^{G_1} = B_i^{G_2}, \forall X_i \in X$. □

Notice that two Markov equivalent classes could entail the same B_X. For example...

Table 1: Comparison between the number of chordal graphs , the number of weak recursively simplicial graphs, the number of undirected graphs and the number of Markov equivalent classes.

# nodes	# chordal graphs	# weak r.s. graphs	# undirected graphs	# Markov equivalent classes
1	1	1	1	1
2	2	2	2	2
3	8	8	8	11
4	61	61	64	185
5	822	882	1024	8782
6	18154		32768	1067825
7	617675		2097152	312510571
8	30888596		268435456	212133402500
9	2192816760		68719476736	326266056291213
10	215488096587 (2×10^{11})		35184372088832 (3×10^{13})	118902054495975141 (1×10^{17})

Corollary 4 $|\{\textit{chordal graphs}\}| \leq |B_X| \leq |\{\textit{Markov equivalent classes}\}|$.

Counting labelled chordal graphs [Wormald, 1985], counting Markov equivalent classes (assymptotic ratio of around 0.27 to DAGs) [Gillispie and Perlman, 2001].

Proposition 8 *Let G be a DAG and F be the moral graph of G. If a node x is a leaf in G, then it must be a simplicial node in F.*

Proof If x is a leaf in G, it has only parents, which form a clique after moralization. By definition, x is a simplicial node in F. □

Corollary 5 *Let G be a DAG and F be the moral graph of G. Then F must have at least one simplicial node.*

Proof Since each DAG has at least one leaf, by Proposition 8 F have at least one simplicial node. □

Corollary 6 *Let G be a DAG and F be the moral graph of G. If a node x is not a simplicial node in F, then it must not be a leaf in G.*

Proposition 9 *Let G be a DAG and F be the moral graph of G. Let S^1 be the set of simplicial nodes in F and F_1 be the induced subgraph of F over $X \setminus S^1$. Then there must exist at least one simplicial node after removing from F' all the edges between $N(X_i), \forall X_i \in S^1$.*

Proof Let F_1' be the result of removing from F_1 all the edges between $N(X_i), \forall X_i \in S^1$. The corresponding directed graph G' of F_1' must be a subgraph of the DAG G, so also acyclic. Assuming F_1' has no simplicial nodes, by Corollary 6 G' has no leaf, which is a contradiction. □

- Simplicial nodes in the first step always contain the leaves.
- Those nodes that become simplicial in the next step without having to delete any edges contain the leaves in the next step.

Here are some issues worth discussing:

1. MBs in DAGs contain parents, children and spouses, whilst MBs in MRFs contain neighbours. Is anyone interested in learning MRFs, who can be oriented into DAGs without changing their MBs?
2. Equivalently, is anyone interested in learning MRFs whose conditional independencies can be represented by DAGs.
3. Some MRFs can be factorized according to the cliques of the graph, if they are chordal graphs (or have positive density by the Hammersley-Clifford theorem, don't know, don't care). If no factorization exists, can create factor graphs. Maybe some non-chordal MRFs can also be factorized by orienting to DAGs as long as they are weak r.s.?
4. Back to MB learning problem: learning MBs of all nodes simultaneously require MBs to be symmetric and have a DAG extension. The symmetric requirement isn't too hard to check during learning, but have a DAG extension is not trivial if the learner only learns MBs, no subgraphs within them. So if Proposition 7 is correct, the MBs learning problem is equivalent as a certain subset of MRF learning.
5. checking if a graph is chordal can be performed in linear time, so checking if a graph is weak r.s. can also be performed in linear time, it will only take a couple of more steps. This could be a quick way of checking the consistency of MBs with DAGs.
6. Once MBs are learned, based on wrs graphs, we have a number of ways of orienting the undirected graph. Depending which simplicial node we go first, one of the wrs node ordering will give the correct DAG. This may have less possibilities than the number of total orderings. Instead of learning the subgraph within each mb, it is easier to get the directions directly.
7. based on how the function f is defined, there are only false positive edges in the wrs graph, no false negatives. But as soon as wrs is oriented, some true positives are mistakenly deleted so false negatives are introduced.
8. we can sample a node ordering and check if the sampling ordering is consistent with the wrs graph. If yes, orient graph, if not resample.
9. several dags give one B_X, which then gives one wrs (moral) graph. Hence DAGs are not necessary for inference. B_X is enough. This seems quite obvious and it doesn't rely on my wrs results at all.

3.3 MB prior for CaMML

DAG prior in camml is calculated from TOM prior by

$$p(G) = \frac{m}{|TOMs(n)|},$$

for a DAG G with n variables and m consistent TOMs. The number of TOMs of n variables can be easily calculated as

$$TOM(n) = n! * 2^{\binom{n}{2}}.$$

TOM prior is assumed uniform in camml, so once we know the total number of TOMs, we know an uninformative TOM prior. Assuming there are m TOMs that are consistent with a DAG G, then G's prior is the sum of these m TOMs' prior. This gives the first part of MML. The second part consists of model parameter and data given such a model, which can be calculated using MML87. The difficulty of calculating m is avoided by MCMC sampling the TOM space then count the number of times each TOM is visited. This is an approximation of m.

Similarly, we can work out the total number of MBs

$$|MB_x| = 2^{(n-1)}.$$

This is the prior for one MB. If we can work out a prior for all MBs, then work out how many combinations of MBs are consistent with G, then we can specify the DAG prior in an alternative way. camml assumes uniform TOM prior which is uninformative. But MB prior can be informative based on learned MBs from data. This could help with camml.

camml samples tom space, so that it approximates the number m of toms that are consistent with G. Since the number of toms is easily calculate and all toms are equal likely, it's easy to calculate the tom prior. Given an estimated m from mcmc, camml obtains an estimated prior of a dag. Log of the mml score of G gives the likelihood of G (with mml estimate of the parameters?). Together with G's estimated prior, we get its posterior. The rest of the grouping steps are not my concern.

The number of B_X is a lot smaller than TOMs. Same as total/partial ordering, each B_X corresponds to a number of DAGs.

Using the same idea, we have the following options:

- assuming the given mbs are correct, then sampling the part (much smaller) of dag space which is consistent with the given mbs to estimate dag posteriro, then select the dag with the highest posterior. pros: the sampled region is much smaller, though still exponential; cons: we don't have perfect mbs...
- several toms are consistent with one dag, but one mb is consistent with several dags, so given (learned) mbs for all nodes, sampling dag space and count the number m of dags that are consistent with the given mbs, then divide the mbs prior by m to get the prior for dag, then using camml steps. the mcmc step can start with a dag that is consistent with mbs (e.g. all mbs are neighbours of targets) then apply mutations (without violating consistent with the given mbs), do muations say 2 million times, afterwards count the number of unique dags.
-

3.4 CaMML* for MB subgraph learning

There are two issues with the current camml. Firstly, the operations for searching the best TOM are not valid for subgraph search within MB, so new operations are needed. Secondly, we don't want to join learned subgraphs into SEC and MMLSEC because we want to stay on DAG level so that we can joint subgraphs into one.

New operations:

1. arc addition: add a directed arc between two variables as long as the resulting graph is still acyclic;
2. arc deletion: selecte a variable, delete one of its (incoming or outgoing) arcs if the variable is directed connected with the target and one of the target's children;
3. swap order: swap the order of two variables except for the target node. If an arc exists between two variables, change the direction after swapping.
4. we need a big step?

These operations ensure the resulting TOM represent a MB subgraph. We know there will be false positives and false negatives in MB discovery. We can leave FNs for now, since although the learned MB is a subset of the true MB, the best subgraph over the learned MB is still a true subgraph of the true model G (proof needed). The FPs are not true MB variables, hence we need to relax step two to also consider deleting arcs so that a variable is outside of the current learned MB. Hence, we define the following step to replace step 2:

2*. arc deletion: selecte a variable, delete (with high probability) one of its (incoming or outgoing) arcs if the variable is directed connected with the target and one of the target's children; alternatively, delete (with low probability) any of its arcs;

In addition, we could use the best MB polytree as a starting point of MCMC sampling. This could save camml some time by not going throug step 1 (annealing step for best TOM for later MCMC sampling) to look for a good starting point.

References

[Friedman and Koller, 2000] N. Friedman and D. Koller. Being Bayesian about network structure. In *Proceedings of the 16th conference on Uncertainty in Artificial Intelligence*, pages 201–210. Morgan Kaufmann Publishers Inc., 2000.

[Friedman *et al.*, 1999] N. Friedman, M. Goldszmidt, and A. J. Wyner. On the application of the bootstrap for computing confidence measures on features of induced Bayesian networks. In *AISTATS*, 1999.

[Gao and Ji, 2017] T. Gao and Q. Ji. Efficient score-based Markov Blanket discovery. *International Journal of Approximate Reasoning*, 80:277–293, 2017.

[Gao *et al.*, 2017] T. Gao, K. Fadnis, and M. Campbell. Local-to-Global Bayesian Network Structure Learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1193–1202, 2017.

[Gillispie and Perlman, 2001] S. B. Gillispie and M. D. Perlman. Enumerating Markov equivalence classes of acyclic digraph models. In *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence*, pages 171–177. Morgan Kaufmann Publishers Inc., 2001.

[Nägele *et al.*, 2007] A. Nägele, M. Dejori, and M. Stetter. Bayesian substructure learning-approximate learning of very large network structures. In *European Conference on Machine Learning*, pages 238–249. Springer, 2007.

[Pellet and Elisseeff, 2008] J. P. Pellet and A. Elisseeff. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9(Jul):1295–1342, 2008.

[Pensar *et al.*, 2014] J. Pensar, H. Nyman, J. Niiranen, and J. Corander. Marginal pseudo-likelihood learning of Markov network structures. *arXiv preprint arXiv:1401.4988*, 2014.

[Riggelsen, 2005] C. Riggelsen. MCMC learning of Bayesian network models by Markov blanket decomposition. In *Preceedings of the 16th European Conference on Machine Learning*, pages 329–340. Springer, 2005.

[Riggelsen, 2008] C. Riggelsen. Learning Bayesian networks: a MAP criterion for joint selection of model structure and parameter. In *Preceedings of the 8th IEEE International Conference on Data Mining*, pages 522–529. IEEE, 2008.

[Wormald, 1985] N. C. Wormald. Counting labelled chordal graphs. *Graphs and combinatorics*, 1(1):193–200, 1985.