Learning Bayesian Networks with Low Rank Conditional Probability Tables

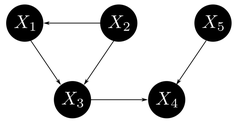
Adarsh Barik, Jean Honorio
abarik@purdue.edu, jhonorio@purdue.edu
Department of Computer Science, Purdue University, West Lafayette - IN, 47907.

OVERVIEW

- Bayesian networks can be used to model complex interactions amongst constituent variables of a system
- **Question:** Can we learn the structure (directed) of a Bayesian network by observing variables of a system?
- **Answer:** NP-hard in general. We propose a method to learn exact structure of a class of Bayesian networks by making black-box queries.
- **Theoretical Guarantees:** We formally prove the correctness of our method and provide polynomial bounds on sample and time complexity.

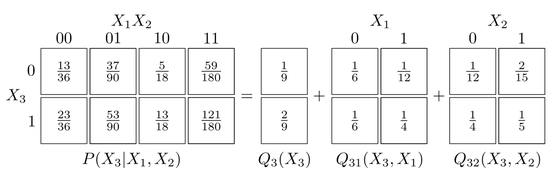
PRELIMINARIES

- **Bayesian network:** A Bayesian network represents a joint probability distribution over the set of random variables defined on the nodes of DAG which factorizes according to the DAG structure, i.e., $P(X_1, X_2, X_3, X_4, X_5) = P(X_1|X_2)P(X_2)P(X_3|X_1, X_2)P(X_4|X_3, X_5)P(X_5)$.



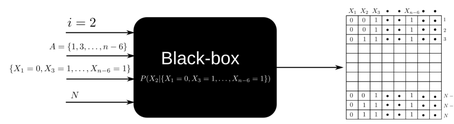
- **Rank k conditional probability distribution** A node i of a Bayesian network is said to be rank k representable with respect to a set $A \subseteq V\backslash\{i\}$ and probability distribution P if $\forall x_i \in \text{dom}(X_i), x_{A(i)} \in \text{dom}(X_{A(i)})$,

$$P(X_i = x_i | X_A = x_A) = \sum_{\substack{S \subseteq \{i\} \cup A \\ 1 \leqslant |S| \leqslant k,\ i \in S}} Q_S(X_S = x_S)$$



Example of Rank-2 CPT

PROBLEM DESCRIPTION



Can we recover the DAG structure using *blackbox queries* with theoretical guarantees of correctness and efficiency in terms of time and sample complexity?

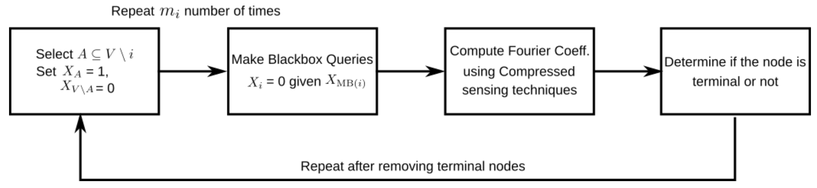
OUR CONTRIBUTION

- *Introduction of Low rank CPTs:* CPT can be treated as summation of multiple simple tables, each of them depending only on a handful of parents
- Connect this notion of rank of a CPT to the Fourier transformation of a specific real valued set function [1]

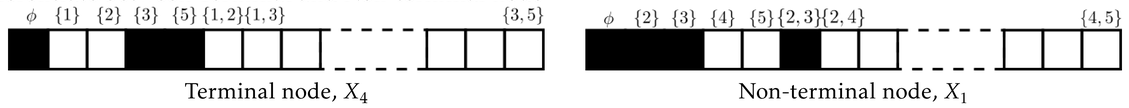
For a set function $f : 2^T \to \mathbb{R}$, $f(A) = \sum_{B \in 2^T} \hat{f}(B)(-1)^{A \cap B}$ where Fourier coefficient $\hat{f}(B) = 2^{-|T|} \sum_{A \in 2^T} f(A)(-1)^{|A \cap B|}$.

- Use compressed sensing techniques to show that the Fourier coefficients of this set function can be used to learn the structure of the Bayesian network

Algorithm



Differentiate between Terminal and Non-terminal node



MAIN THEOREM

Let $g_i(A_j)$ be an approximation for $f_i(A_j) = P(X_i = 0|X_{\text{MB}(i)})$ by taking only upto pairwise Fourier coefficients. Let $|f_i(A_j) - g_i(A_j)| \leqslant \epsilon_j, \forall A_j \in \mathcal{A}_i$ for some $\epsilon_j > 0$. $\hat{\boldsymbol{g}}_{\boldsymbol{i}}$ by solving the following optimization problem for each node i.

$$\boldsymbol{\beta}_i = \min_{\hat{\boldsymbol{g}}_{\boldsymbol{i}} \in \mathbb{R}^{|\rho_i|}} \|\hat{\boldsymbol{g}}_{\boldsymbol{i}}\|_1 \quad s.t. \|\mathcal{M}_i \hat{\boldsymbol{g}}_{\boldsymbol{i}} - \boldsymbol{f}_{\boldsymbol{i}}\|_2 \leqslant \epsilon \quad \text{where } \epsilon = \sqrt{\sum_{A_j \in \mathcal{A}_j} \epsilon_j^2}\,. \tag{1}$$

Let $\mathcal{A}_i$ be a collection of m_i sets $A_j \in 2^{V-i}$ chosen uniformly at random and $\boldsymbol{g}_{\boldsymbol{i}} \in \mathbb{R}^{m_i}$ be a vector whose j^{th} row is $g_i(A_j)$ and $\hat{\boldsymbol{g}}_{\boldsymbol{i}} \in \mathbb{R}^{n+\binom{n-1}{2}}$ be a vector with elements of form $\hat{f}_i(B_k) \forall B_k \in \rho_i$ where $\rho_i = \{B_k \mid B_k \in 2^{\bar{i}}, |B_k| \leqslant 2\}$ is a set which contains support$(\hat{f}_i)$. Consider, $\boldsymbol{g}_{\boldsymbol{i}} = \mathcal{M}_i \hat{\boldsymbol{g}}_{\boldsymbol{i}}$ where, $\mathcal{M}_i \in \{-1, 1\}^{m_i \times n}$ such that $\mathcal{M}^i_{jk} = (-1)^{|A_j \cap B_k|}$.

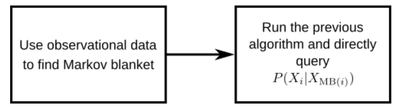
Theorem 1. *Suppose $\hat{\boldsymbol{g}}_{\boldsymbol{i}}$ is constructed by computing $\hat{g}_i(B_k)$ using B_k from a fixed collection ρ_i and $\boldsymbol{g}_{\boldsymbol{i}}$ is computed by selecting m_i sets A_j uniformly at random from $2^{\bar{i}}$. Let $m_i \geqslant \mathcal{O}(\max(|support(\hat{g}_i)| \log^4(n + \binom{n-1}{2}), |support(\hat{g}_i)| \log \frac{1}{\delta})$ and $\boldsymbol{\beta}_{\boldsymbol{i}}$ be solved using equation (1). Then with probability at least $1 - \delta$, we have $\|\boldsymbol{\beta}_{\boldsymbol{i}} - \hat{\boldsymbol{g}}_{\boldsymbol{i}}\|_2 = \mathcal{O}(\frac{\epsilon}{\sqrt{m_i}})$. If the minimum non-zero element of $|\hat{\boldsymbol{g}}_{\boldsymbol{i}}|$ is $\Omega(\frac{\epsilon}{\sqrt{m_i}})$ then $\boldsymbol{\beta}_{\boldsymbol{i}}$ recovers $\hat{\boldsymbol{g}}_{\boldsymbol{i}}$ up to the signs. Furthermore, if non-terminal nodes are not rank 2 then $|\boldsymbol{\beta}_{\boldsymbol{i}}(B)| = \mathcal{O}(\frac{\epsilon}{\sqrt{m_i}}), \forall B \in \rho_i, |B| = 2$ iff i is a terminal node and $\hat{\pi}(i) = \{B \mid |B| = 1,\ |\boldsymbol{\beta}_{\boldsymbol{i}}(B)| = \Omega(\frac{\epsilon}{\sqrt{m_i}})\}$ correctly recovers the parents of node i.*

RESULTS

Algorithms	Sample Complexity	Time Complexity	Selections	Queries
Our Work (no observational data)	Blackbox - $\mathcal{O}(nk^3 \log^4 n(\log k + \log\log n))$	$\mathcal{O}(n^4 k\sqrt{n} \log n)$	$\mathcal{O}(n)$	$\mathcal{O}(nk^3 \log^4 n)$
Our Work (with observational data)	Observational - $\mathcal{O}(n)$, Blackbox - $\mathcal{O}(nk^3 \log^5 k)$	$\mathcal{O}(n^4)$, $\mathcal{O}(nk^4\sqrt{k} \log k)$	$\mathcal{O}(n)$	$\mathcal{O}(nk^3 \log^4 k)$
Bello et. al, 2018	Interventional - $\mathcal{O}(n^2 2^k \log n)$	$\mathcal{O}(n^2 2^k \log n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2 2^k)$
Kocaoglu et. al, 2017	Interventional - no guarantees	$\mathcal{O}(2^n kn^2 \log^2 n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(2^n \log n)$

Here n is the number of nodes, k is the maximum size of the Markov blanket. The maximum number of parents of a node is $\mathcal{O}(k)$.

USING OBSERVATIONAL DATA

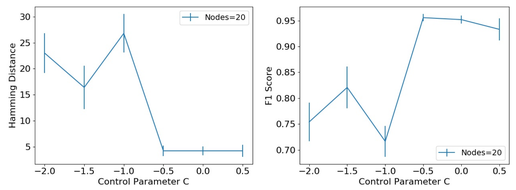


Let $\mathbb{P} = \{P \text{ is faithful to } G.\ |P(X_i|X_l) = P^*(X_i|X_l), \forall i, l \in \{1, \ldots, n\}\}$. If there exists a probability distribution $\hat{P} \in \mathbb{P}$ such that each node i is rank 2 with respect to $\text{MB}_G(i)$ and $\hat{P}$, then we can recover the Markov blanket by solving the following system of equations:

$$P(X_i = 0, X_l = 0) = \tilde{Q}_i(X_i = 0)P(X_l = 0) + \sum_{\substack{j \in -i \\ j \neq l}} \tilde{Q}_{ij}(X_i = 0, X_j = 0)P(X_j = 0, X_l = 0) + \tilde{Q}_{il}(X_i = 0, X_l = 0)P(X_l = 0) \forall\ l = \{1, \ldots, n\}, l \neq i$$

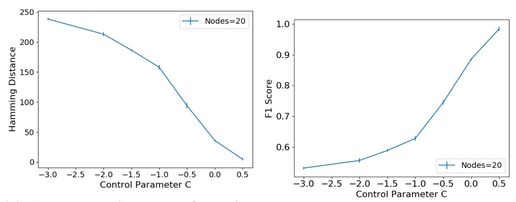
$$P(X_i = 0) = \tilde{Q}_i(X_i = 0) + \sum_{\substack{j \in -i \\ j \neq l}} \tilde{Q}_{ij}(X_i = 0, X_j = 0)P(X_j = 0)$$

VERIFICATION



(a) Hamming distance with control parameter C (b) $F1$ score with control parameter C

Figure 1: Without observational data. $m_i = 10^C \max(k^2 \log^4 n', k^2 \log 1/\delta)$ queries for each node i.



(a) Hamming distance of Markov blanket recovery with control parameter C (b) $F1$ score of Markov blanket recovery with control parameter C

Figure 2: With observational data. $N = \max(10^C \frac{\log n}{\epsilon^2}, n))$ observational samples

References

[1] Peter Stobbe and Andreas Krause. Learning Fourier Sparse Set Functions. In *Artificial Intelligence and Statistics*, pages 1125–1133, 2012.