J. Parallel Distrib. Comput. 73 (2013) 1039–1048

Contents lists available at SciVerse ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Parallel globally optimal structure learning of Bayesian networks

Olga Nikolova[a], Jaroslaw Zola[b], Srinivas Aluru[c,d,*]

[a] Sage Bionetworks, Seattle, WA 98109, USA

[b] Rutgers Discovery Informatics Institute, Rutgers University, Piscataway, NJ 08854, USA

[c] Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

[d] Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India

HIGHLIGHTS

- First parallel algorithm for learning optimal structure of Bayesian networks.
- Work optimal, with space complexity within a factor of 1.41 of the optimal.
- First investigation of Bayesian network learning with bounded node in-degree d.
- Proof that $d < n/3 - \log mn$ can be learned as efficiently as $d = O(1)$.
- Extensive experimental validation on Blue Gene/P and Opteron InfiniBand cluster.

ARTICLE INFO

Article history:
Received 4 January 2012
Received in revised form
1 April 2013
Accepted 2 April 2013
Available online 12 April 2013

Keywords:
Bayesian networks
Graphical models
Machine learning
Structure learning
Parallel algorithm

ABSTRACT

Given n random variables and a set of m observations of each of the n variables, the Bayesian network structure learning problem is to learn a directed acyclic graph (DAG) on the n variables such that the implied joint probability distribution best explains the set of observations. Bayesian networks are widely used in many fields including data mining and computational biology. Globally optimal (exact) structure learning of Bayesian networks takes $O(n^2 \cdot 2^n)$ time plus the cost of $O(n \cdot 2^n)$ evaluations of an application-specific scoring function whose run-time is at least linear in m. In this paper, we present a parallel algorithm for exact structure learning of a Bayesian network that is communication-efficient and work-optimal up to $O\left(\frac{1}{n} \cdot 2^n\right)$ processors. We further extend this algorithm to the important restricted case of structure learning with bounded node in-degree and investigate the performance gains achievable because of limiting node in-degree. We demonstrate the applicability of our method by implementation on an IBM Blue Gene/P system and an AMD Opteron InfiniBand cluster and present experimental results that characterize run-time behavior with respect to the number of variables, number of observations, and the bound on in-degree.

1. Introduction

1.1. Bayesian networks and Bayesian structure learning

Bayesian networks (BNs) are graphical models which represent probabilistic relationships among interacting random variables of a given domain. In BNs, relationships between variables are depicted in the manner of conditional independences and quantitatively assessed by conditional probability distributions [9]. BNs provide a holistic view of the interactions within a domain and can be learned from data. They have been successfully applied in medical and fault diagnosis, bioinformatics, e-commerce, user preference prediction, spam filtering, etc., and have been widely used in recent decades [2,6,11,12,24].

Formally, a Bayesian network is defined to consist of two components. Let P specify the joint probability distribution over some set of random variables $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$, and let $N = (\mathcal{X}, E)$ be a directed acyclic graph (DAG). In N, a node X_j is called a *parent* of X_i if an edge from X_j to X_i exists. A node X_k is called a *descendant* of X_i if there is a directed path from X_i to X_k and is termed a *nondescendant* of X_i if no such path exists. The pair (N, P) defines a Bayesian network if each variable in $\mathcal{X}$ is independent of its nondescendants, given its parents, denoted by $I_P(X_i, ND(X_i) \mid Pa(X_i))$, where $ND(X_i)$ and $Pa(X_i)$ denote the nondescendants and parents in N of X_i, respectively. This condition is known as the Markov assumption [16]. Given that (N, P) satisfies the Markov assumption, and from the chain rule of probability, P

[*] Corresponding author at: Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA.

E-mail addresses: olga.nikolova@sagebase.org (O. Nikolova), jaroslaw.zola@rutgers.edu (J. Zola), aluru@iastate.edu (S. Aluru).

0743-7315/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jpdc.2013.04.001

is decomposable in the following product form:

$$P\left(X_1, X_2, \ldots, X_n\right) = \prod_{i=1}^{n} P\left(X_i \mid Pa\left(X_i\right)\right).$$

When learning the structure of a BN from data, the Bayesian approach utilizes a statistically motivated scoring function to evaluate the posterior probability of a considered graph, given the data [4,10]. One example of such a scoring function is the Bayesian score. Suppose we are given m observations of the system, each observation consisting of the value of each of the n random variables in $\mathcal{X}$. These observations can be viewed as an $n \times m$ matrix $D_{n\times m}$, with rows representing variables and columns representing observations. Then, the Bayesian score for a network N given $D_{n\times m}$ is

$$\text{Score}\left(N\right) = \log P\left(N \mid D\right) = \log P\left(D \mid N\right) + \log P\left(N\right) + C,$$

where C is a constant. In this score the two non-constant terms $\log P\left(D \mid N\right)$ and $\log P\left(N\right)$ refer respectively to the log-likelihood of D given N and the prior probability of N, usually taken to be uniform.

To find the optimal network efficiently, it is crucial to choose a scoring function which decomposes into individual score contributions $s\left(X_i, Pa\left(X_i\right)\right)$ of each of the variables in $\mathcal{X}$ given its parents:

$$\text{Score}\left(N\right) = \sum_{X_i \in \mathcal{X}} s\left(X_i, Pa\left(X_i\right)\right).$$

Examples of Bayesian and information theory scoring functions include Bayesian Dirichlet (BD) [4], Bayesian Information Criterion (BIC) [20], and Minimum Description Length (MDL) scoring function [14]. More generally, such scoring functions are shown to, with high probability, assign superior scores to graphs which more precisely depict the dependences in the data [9]. BNs are said to be *equivalent*, and therefore indistinguishable, if they represent the same set of independences [8]. Finally, optimizing the scoring criterion facilitates the discovery of equivalent structures that best represent the observed data.

A major difficulty in BN structure learning is the super exponential search space in the number of random variables. As reported by Robinson [19], for a set of n variables there exist $\frac{n!2^{\frac{n}{2}(n-1)}}{r\cdot z^n}$ possible DAGs, where $r \approx 0.57436$ and $z \approx 1.4881$. To reduce the search space, marginalization over node orders has been proposed [13,18,21]. Here we briefly review the main idea.

Consider the graph $N = (\mathcal{X}, E)$ of a BN (N, P). Due to its DAG structure, the random variables in $\mathcal{X}$ can be ordered in such a way that for each variable $X_i \in \mathcal{X}$ its parents $Pa\left(X_i\right)$ precede it in the specified ordering. Optimizing a scoring function for each variable $X_i \in \mathcal{X}$ and all of its respective subsets of preceding elements allows us to discover the optimal network that respects the particular ordering. Further, to investigate all possible graphs we need to consider a total of $n!$ orderings of $\mathcal{X}$. This yields the search space of ordered subsets of $\mathcal{X}$ of size $n!2^n$. However, as long as $Pa\left(X_i\right)$ precedes X_i, the order among the nodes in $Pa(X_i)$ is irrelevant. Thus, the search space can be reduced to the 2^n unordered subsets of $\mathcal{X}$.

Even reduced, the search space remains exponential and in practice, exact structure learning without additional constraints has been achieved only for moderate size domains and at high execution cost. While exact network structure learning, or globally optimal structure learning, has been shown to be NP-hard even with bounded node in-degree [3], constructing exact Bayesian networks without additional assumptions remains valuable nevertheless.

1.2. Contributions

In this paper, we present a parallel algorithm for exact Bayesian network structure learning with optimal parallel efficiency on up to $O\left(\frac{1}{n}\cdot 2^n\right)$ processors regardless of the complexity of the scoring function used. We also investigate structure learning with user-specified bounded in-degree d, which is the maximum allowed in-degree (or equivalently, the number of parents) of any node in the network. We show that for $d < \frac{1}{3}n - \log mn$ the asymptotic run-time complexity remains unaffected as a function of d. An important consequence of this result is that networks with a generous allowance of bounded in-degree can be learnt in the same time as networks where only constant in-degree is allowed. We also show that $d \geq \lceil \frac{n}{2} \rceil$ results in the same asymptotic run-time complexity as $d = n - 1$, where no restriction is placed on the in-degree. To assess the performance of our algorithms experimentally, we report results on two parallel systems—IBM Blue Gene/P and AMD Opteron cluster with InfiniBand interconnect. Experimental results demonstrate the validity of the theoretical predictions.

1.3. Related work

In the past, parallel algorithms have been developed for heuristics-based Bayesian network learning methods, particularly using meta-heuristics and sampling techniques [1,15]. Heuristics-based algorithms trade off optimality for the ability to learn larger networks. The presented work in exact Bayesian learning is intended to push the scale of networks for which optimal networks can be inferred. Our parallel algorithm is based on Ott et al.'s sequential algorithm [18] that takes $O(n2^n)$ steps and has a run-time and space complexity of $O(n^2 2^n)$. To our knowledge, the presented work is the first parallel algorithm for exact learning, with an initial conference version published in 2009 [17]. In addition, the bounds established in this paper for run-time as a function of the maximum node in-degree are new both in sequential and parallel settings. In 2011, Tamada et al. [22] present a different parallel algorithm for globally optimal structure learning, also based on Ott et al.'s sequential algorithm. Because of the shared objective and as both parallel algorithms are based on the same sequential algorithm, we review the work in more detail here.

Tamada et al.'s [22] parallel algorithm is based on an entirely different strategy than the one presented in this paper. Both approaches compute optimal networks for all 2^n subsets of $\mathcal{X}$ and take advantage of the structure of optimal solutions through dynamic programming as in Ott et al. [18]. Tamada et al. compute the optimal networks for all subsets of the same size in parallel. They exploit the observation that computation on a subset can be flexibly structured based on availability of computation on smaller subsets contained within it, and communication savings can be realized by scheduling the same size subsets with significant overlaps on the same processor. To reduce communication, the algorithm presents a trade-off where calculations are sometimes performed redundantly, thus increasing both overall work complexity and space complexity of the parallel algorithm. As in Ottet al., they present run-time and space complexities as the number of steps, while we prefer to report run-time and space complexities in terms of actual time and space. Multiplication by $O(n)$ provides for correct translation from the number of steps to actual complexities. Tamada et al.'s algorithm has a work complexity of $O(n^{\sigma+1}2^n)$ steps, which translates to a work and space complexity of $O(n^{\sigma+2}2^n)$, for any integer $\sigma \geq 1$. Thus, the algorithm performs $O(n^\sigma)$ more work and takes $O(n^\sigma)$ more space asymptotically over the