学 位 論 文 内 容 の 要 旨

博士の専攻分野の名称　博士（情報科学）　氏名 高 徯

学 位 論 文 題 名

Fast Exact Inference Algorithms for Bayesian Networks Based on ZDD Operations
（ZDD 演算に基づくベイジアンネットワークの高速かつ厳密な確率推論アルゴリズム）

Compiling Bayesian networks (BNs) into secondary structures to implement efficient exact inference is a hot topic in probabilistic modeling. One class of algorithms to compile BNs is to transform the BNs into Zero-Suppressed Binary Decision Diagrams (ZDDs) to perform efficient exact inference. This method has attracted much attention. A ZDD is a data structure for manipulating boolean functions and item combinations efficiently. By compiling a BN into ZDDs, computation time for exact inference using ZDDs is reduced to linear time in the size of the ZDDs. Also, cache memory techniques further help to accelerate the inference. However, as the size of BN grows, compiling ZDDs becomes unacceptable in both time consumption and ZDD size which hinders BN practical applications. In this thesis, we focus on improving the ZDD-based method to efficiently execute the exact inference of BNs. We take into account two aspects, condensing ZDD size through factorizations and compiling the decomposition form of BNs instead of the whole network.

In Chapter 3, to condense ZDD size, we propose a fast factorization method based on the d-separation structures in BNs to factor the ZDDs with large size into small ones. The weak divison algorithm is used to factorize a large sum-of-product into several compact sum-of-products. It is known as the most successful and prevalent technique of logic synthesis and optimization. Minato et al. proposed an improvement of this algorithm, known as the fast weak division algorithm for ZDD-based logic operations. In this algorithm, variables appearing many times are iteratively extracted and then used as divisors to factorize a ZDD.We can use this algorithm to factorize a large ZDD into small ones. However, for a ZDD representing a BN, the approach to use every multiple appearing variables as divisors to factorize a ZDD leads to unacceptable time consumption for factorization. We improve this factorization algorithm by extracting divisors using d-separations in BNs. In our method, variables appearing multiple times are extracted once so that time consumption are largely reduced. What is more, the resulting ZDD is largely condensed which would result in big improvements in time of exact inference.

In Chapter 4, we propose a fast message passing algorithm using ZDD-based local structure compilation. The idea of factorizing ZDDs is based on the fact that we are able to generate ZDDs for a BN. As BN gets large, the size of resulting ZDD will be too large to generate at the first place. Therefore, we consider to compile the decomposition form of a BN instead of a whole BN into ZDDs. A junction tree is one of the most prevalent decomposition forms of a BN whose node is a clique consisting of BN vertexes. Performing the message passing algorithm on a junction tree for exact inference is currently one of the most prominent BN inference algorithms. The algorithm works by passing real valued functions called messages along with the edges between the two nodes in a junction tree. The performance

of this algorithm depends on a BN's treewidth or the optimal maximal clique size of a corresponding junction tree. In our method, a junction tree is directly compiled into ZDDs. We introduce message variables in ZDDs to pass messages. Then, the message passing algorithm can be performed on ZDDs. By utilizing techniques of node sharing and cache memory in ZDDs, parameters in BNs which share the same value can compactly represented. Moreover, repetitive local computations during the message passing algorithm are avoided. Our method of conducting message passing on ZDDs performs much faster than the performing on the original junction tree which is generated through a well known heuristic way called min-fill method.

In Chapter 5, we present the method of separate compilation of BNs for efficient exact inference. We propose to combine these two approaches that using the d-separation structure in BNs to partition a large BN into several components. For every given BN, serial pattern d-separation sets are found and used to partition the BN into conditionally independent components. Then we compile these components into small ZDDs and perform exact inference using these ZDDs. Separately compiling these components into ZDDs is more efficient than generating a giant ZDD for a whole network. However, partitioning a BN into too many components may give rise to considerable time consumption which grows exponentially with the number of vertexes in serial pattern d-separations. To trade off the off-line time consumption (for finding d-separations and compiling ZDDs) and on-line time consumption (for inference using ZDDs), the d-separations used to partitioning BNs are restricted to one-vertex and found using Tarjan's vertex-cut algorithm which can be performed linear time in the number of BN vertexes. The experiments illustrate that one-vertex d-separations exist in most BNs. Partitioning BNs with one-vertex d-separations improves the speed for both compilation and inference significantly than the conventional ZDD-based method.

In Chapter 6, we conclude our remarks and discuss the future work and open problems. We show that ideas in this thesis are valuable since not only for ZDDs, they are also usable for other data structures. We hope to improve other logic operation based approaches such as d-DNNFs, an efficient logic circuit used in BN inference recently. We expect that using fast logic operations can bring a big improvement for the exact inference of BNs.