Hybrid Bayesian Network Structure Learning Using Map-Reduce For Big Data

By

Jarrar Haider

Fall 2017-MS(CS-07)-00000202973

Supervisor

Dr. Sohail Iqbal

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MS CS)

In

School of Electrical Engineering and Computer Science,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(July 2021)

Approval

It is certified that the contents and form of the thesis entitled "**Hybrid Bayesian Network Structure Learning Using Map-Reduce For Big Data**" submitted by **Jarrar Haider** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Sohail Iqbal**

Signature: ____________________

Date: ____________________

Committee Member 1: **Dr. Amanullah Yasin**

Signature: ____________________

Date: ____________________

Committee Member 2: **Dr. Safdar Abbas Khan**

Signature: ____________________

Date: ____________________

Committee Member 3: **Dr. Asad Waqar Malik**

Signature: ____________________

Date: ____________________

Thesis Acceptance Certificate

Certified that final copy of MS thesis written by Ms **Jarrar Haider** (Registration No **Fall 2017-MS(CS-07)-00000202973**), of SEECS has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Advisor: **Dr. Sohail Iqbal**

Signature: ________________

Date: ________________

Head of Department (HoD):

Signature: ________________

Date: ________________

Dean/Principal:

Signature: ________________

Date: ________________

Dedication

Dedicated to my parents, who have been a constant support even when I was down and out

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Jarrar Haider**

Signature: ________________

Acknowledgment

All gratitude is to Almighty Allah, the source of knowledge and wisdom, the most Gracious, the most Merciful, who blessed me with the acumen to complete this thesis successfully.

I would like to express my deepest gratitude to Dr. Sohaik Iqbal, my thesis supervisor, and Dr. Amanullah Yasin, my thesis co-supervisor, for their continuous support and motivation. None of this would have been possible without their perceptiveness, motivation and command on the subject.

My appreciation goes to my committee members who monitored my work and took effort in reading and providing valuable comments on my presentations and other thesis documents.

I would also like to acknowledge my family and friends who have always been there to cheer and support me. Their words of encouragement and appreciation have always inspired me to keep going in life.

Contents

List of Abbreviations

MMHC	Max-Min Hill Climbing
PC	Parent Children
HC	Hill Climbing
MMPC	Max-Min Parents and Children
CPT	Conditional Probability Tables
DAG	Directed Acyclic Graphs
CI	Conditional Independence
BN	Bayesian Network
MR	MapReduce

List of Figures

Abstract

Predicting the future or outcome of any event has been in human's nature since the beginning of time. We have always been trying to know what our actions will result in. With the advancement in the field of science and technology and with the increase in processing power of computer hardware, we have come very close to predicting certain outcomes based on prior knowledge. Bayesian networks are one the ways of predicting an outcome. It falls into the category of Probabilistic Graphical Model. It finds it use in data mining and for representing uncertain knowledge. Big data, artificial intelligence and machine learning rely on data and gets effected by changes in it. Bayesian Network helps in understanding the data and finding meaningful inferences, which are often basis of realistic applications. In this paper, we are going to discuss how Max-Min Hill Climbing, that is a hybrid algorithm, with Map-Reduce based framework can be implemented in order to lessen the execution time with similar accuracy.

Chapter 1

Introduction

1.1 What is Bayesian Network

Bayesian Networks is often considered as an important choice for prediction, classification and analysis of unknown events, knowledge and factors. There are two different learning that are involved in a Bayesian network. Directed Acyclic Graphs (DAG) are used to show dependent or independent relationships between the variables whereas Conditional Probability Tables (CPT) is used to determine the strength between DAG. Finding a DAG is categorized as a Structure Learning whereas finding relations between variables using conditional probability tables falls into the category of Parameter Learning. Finding of a DAG is of more interest to researchers as structure learning is a NP-hard problem.

1.1.1 Bayesian Network Classification

Graphical representation illustrated below helps us to understand the classification of different learning methods and techniques involved in Bayesian methods. (Figure 5.2)

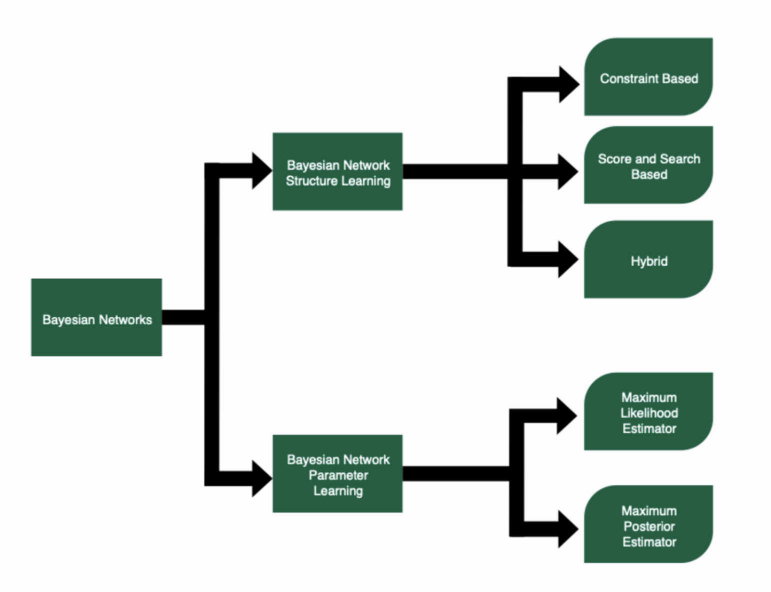
Bayesian Networks
Bayesian Network Structure Learning
Bayesian Network Parameter Learning
Constraint Based
Score and Search Based
Hybrid
Maximum Likelihood Estimator
Maximum Posterior Estimator

Figure 1.1: Bayesian Network Classification

From looking at (Figure 5.2), Bayesian network learning is mainly divided into structured learning and parameter learning. In this paper, we will discuss hybrid bayesian network structure learning and how it can be implemented using Map-Reduce parallel learning technique to reduce over all execution time and achieve similar accuracy to traditional sequential approaches.

1.1.2 Bayesian Network Structure Learning

Bayesian network structure learning can be explained as a learning technique in which Directed Acyclic Graphs (DAG) are used to show dependent or independent relationships between the variables. They are further categorized into three different learning approaches.

1.1.2.1 Constraint Based

Constraint based learning method tends to form a graph structure that shows relationship between variables for given data and find dependence and independence between them, that matches empirical distribution. There are number of different algorithms that fall under the category of constraint based bayesian network learning approach. Some of the examples are:

- PC Algorithm
- Grow-Shrink (GS)
- Markov Blanket
- Max-Min Parents Children (MMPC)

1.1.2.2 Score and Search Based

As evident from the name, score-and-search based approach consists of two parts score based approach and search based approach.

Score based approach follows the methodology to evaluate when data is fitted for a Bayesian Network. Evaluation is actually a scoring function that assigns scores to variables

Search based approach is the used to find the maximum score over a given DAG space.

Score Metric can be defined as $\text{Score}\ (G|D) = LL(G:D) - \emptyset(|D|)\ ||G||$

Here G refers to structure, D refers to data, LL(G:D) refers to log-likelihood of the data under the graph structure G

Some of the examples of score-and-search based algorithms are:

- Hill Climbing (HC)
- Tabu Search (Tabu)

1.1.2.3 Hybrid based approach

As the name suggests, hybrid based approach is a combination of constraint based and score-and-search based approaches. This is a relatively new approach and only a handful of algorithms are available. General concept of this approach is that first dependence and independence between variables is calculated. This creates a set having edges, that helps in better DAG creation. Once we have a set, we then use score-and-search based approach which gives the maximum score over relationships created in given set and then searches over them to create best possible DAG.

Some of the examples of score-and-search based algorithms are:

- Max-Min Hill Climbing (MMHC)
- Hybrid HPC (H2PC)
- General 2-Phase Restricted Maximization (RSMAX2)

1.1.3 Applications of Bayesian Networks

Bayesian networks have a wide application and are being used from medicine to image processing and spam filtering. They have been providing useful decision making outcomes that helps in better choices. Some of the major application ares for bayesian network are listed below:

- Document Classification
- Semantic Search
- Spam Filtering
- Gene Regulatory Monitoring
- Medicine
- Information Retrieval
- Image Processing
- Turbo Code
- Bio-monitoring

1.2 Motivation

Making informed decisions has always been in favour of humans as it helps them survive, save lives or make money. We have been collecting huge data to make informed decisions but the data is so big and in raw form that it is hard to make something meaningful out of it. By using modern algorithms and techniques, we can feed these large data-sets and get a better understanding and better outcomes that helps us in making an informed decision. In past, Bayesian networks have found its application in the field of medicine and in a country like Pakistan where amount of patients is large and doctors are not that readily available, quick and accurate prediction of diseases based on symptoms can help increase efficiency and reduce chances of errors. Using hybrid bayesian network learning techniques, we are able to find similar results as obtained in non-hybrid approaches with better execution time.

1.3 Objectives of our Research

With the advancement in hardware technology and with modern parallelism techniques that can be used to get the maximum output, we want to establish an approach where hybrid bayesian network structure learning can be used for learning the unknown. To save execution time of any linear approach, we have used map reduced based approach for parallel execution that saves us execution time. Using Max-Min Hill Climbing algorithm, that is hybrid in nature, we achieve similar accuracy as that of constraint based approach but execution times at par with score-and-search based approach and we get best of both approaches with some compromises.

1.4 Problem Statement

Despite the availability of powerful hardware devices with parallel execution capabilities, there is little to no work done in the field of hybrid bayesian network structure learning and exploiting parallelism techniques to achieve similar results with better execution time, especially for big data having multiple variables that tends to take time in giving meaningful results

1.5 Solution Statement

Implementation of Bayesian Network Structure Learning using Hybrid algorithm and distributive learning methodology (Hadoop and MapReduce), for big data (having high number of records and variables), to reduce the execution time and achieve similar accuracy to traditional and sequential structure learning algorithm

1.6 Objective and Research Methodology

With the focus on parallel implementation of a hybrid bayesian network for big data, this research aims at developing a MapReduce based solution that achieves parallel implementation of Max-Min Hill Climbing for similar results with better execution timing then a sequential based approach. Three phase methodology has been adopted to achieve the research objective for this thesis.

The three phases are explained below:

1.6.1 First Phase: Selection of Hybrid BN Algorithm

For research objective of this thesis we have selected Max-Min Hill Climbing (MMHC) algorithm that falls into the category of hybrid bayesian network structure learning.

1.6.2 Second Phase: Implementation of Hybrid Bayesian Network Algorithm using MapReduce

In second phase, following tasks are to be performed:

- Creating MapReduce based environment.
- Implementing MMHC over distributed MapReduce based environment.

1.6.3 Third Phase: Find and compare results with sequential hybrid based algorithms

Third and final phase is as follow:

- Finding the results obtained from running algorithm over MapReduce
- Comparison of results obtained from sequentially running MMHC and measuring accuracy and execution time.

1.7 Thesis Organization

This thesis report has been organized into six main chapters. Chapter one presents introduction of our research topic by describing our motivation, problem statement, solution statement, and objective and research methodology. Chapter two summarizes background study and reviews relevant literature work. Chapter three highlights the research problem and problem solution. Detailed implementation of the algorithm along with important parts are explained in chapter four. Chapter five contains experimental results and discussion whereas chapter six concludes this thesis report by giving directions for future work.

Chapter 2

Literature Review

Bayesian Network structure learning is most important part of Bayesian network learning as DAG gives dependency and independence relationships between variables. The main idea behind structure learning is to select one suitable graph from several possibilities that fits the given data most accurately and gives the best results out of all the possible DAGs. In literature we find that it becomes very difficult to find the most accurate graphical representation as the number of variables increase. Moreover, with the increase in number of variables, the search space also increases making it an expensive process for finding a DAG. To solve this problem more efficiently, researchers have implemented many techniques and algorithms to get the best outcome in least amount of time.

2.1 Constraint based learning

Constraint based learning is one of the techniques that is used for learning the Bayesian network and is used for finding a DAG for a Bayesian network. It follows statistical tests that is used to determine the relation between parents, children and its neighbors. Series of conditional hypothesis tests are performed to find and learn the conditional independence between variables. From these constraints, a directed acyclic graph is learned. All this is based on the hypothesis that is formed based on conditional independence tests.

One of the most popular approaches for learning of constraints is PC algorithms (named after the authors Peter and Clark). According to this algorithm, structure is stared from undirected graph on which recursive independence test is performed between variables. This deletes the edges and we are left with a smaller undirected graph. Symmetry is then learned from this undirected graph and graph can be converted from undirectional to partial directional to a directed acyclic graph.

Anders L. Madsen et al. [5] presented two different approaches to parallelization of conditional independence tests. Aim of this is to reduce the time required in performing these tests as this is the most time-consuming step in finding constraints. Speedups are shown for both, shared memory systems and cluster systems.

With the advancements in technology and methods, parallel computing and processing is now becoming a popular way for solving problems more efficiently. Marco Scutari [17] shows that backtracking technique can be replaced with parallelization of constraint- based algorithms. This can be achieved by combination of software architecture and framework on which the solution is implemented in such a way that it can run in parallel on multiple processors in contrary to single-processor machines.

2.2 Score and search-based learning

Score and Search based learning is one of the most researched topics in Bayesian network learning. As the name suggests, this type of learning has two parts; a score function that is used to find out how well a given data is mapped on the network and a search function that is used to find out which network produces the best score by looking for parents for each variable. As score-and-search based learning is a NP-Hard problem, therefore approximation and greedy algorithms are usually used to find the answer.

Jos é A. G amez et.[1] uses a variant of Hill Climbing Algorithm, that is one of the most common score and search approaches that are followed and improves it by reducing the overall time for learning the network with retaining similar accuracy. It shows a technique of local score-and-search based approach that is better for dealing with larger datasets having large variables. Results show significant improvement in time as compared to traditional Hill Climbing technique.
Attempts have been made in lowering the overall execution time of greedy search algorithm. [12] shows that the speed of Bayesian network learning can be improved by taking advantage of the availability of closed form estimators for local distributions with few parents. Also, it is found that by using

predictive instead of in-sample goodness-of-fit scores helps in improving both speed and accuracy at the same time.

Approximation techniques are utilized in finding the DAG for a Bayesian network and Ordering Based Search (OBS) is one of the approximation algorithms. [8] talks about improvement of OBS by sampling more effectively the space of the orders. To find the quality of the Bayesian Network's Directed Acyclic Graph structure, Bayesian Information Criterion (BIC) has been adopted which is almost proportional to the posterior probability of the DAG. Three things have been explained for optimization: starting from parent set identification, then structure optimization and at the end structure optimization under bounded trees. Better results have been found on large and very large data set.

With the advancement in technology, a lot of research carried out now involves parallel learning approaches. [2] shows parallel and incremental approach of bayesian learning for large scale and changing data, that is distributed and not stored at one place. A classical Hill Climbing algorithm is selected and is run using MapReduce approach for parallelism. Minimum Description Length is being used a scoring metric and the resultant is passed to MapReduce based algorithm that is then used to calculate marginal probabilities of the data. Hill Climbing is then used to get structure and a key-value pair relation. For continuously changing data, a concept of influence degree is introduced. It is used for comparison between old and new data. Confidence of existing data and new data is evaluated using it. Proposed method is both, scalable and effective and it proved by both theoretical and experimental results.

K2 algorithm, which is a traditional score and search-based algorithm. [4] has selected it and has extended it by introducing MapReduce technique for processing massive data in parallel. For finding and learning the structure of the Bayesian network, the process is divided into two parts. First one is the scoring step this is for finding out the required parameters. This is done in parallel over Map Reduce. Second step the searching part which is also implemented over Map Reduce. In the Map part, algorithm is run over every node to find the local optima in parallel and then the resultant local optima structure is selected that has the highest value.

The Reducer then merges all the local optima into global optima. This approach is a way of finding a way to fit massive data using MapReduce in K2 algorithm as traditionally, it is not suitable for large scale massive data. Hidden nodes and large dataset sizes is a problem in Bayesian Networks. [7] uses an extension of parallel Bayesian algorithm called Expectation Maximization (EM) is used to solve the problems mentioned above. It is extended from traditional execution to Map Reduced based execution. Expectation phase of sequential EM consists of two steps. First one is computing marginals using belief propagation and the second is calculating of pseudo-counts for all input data. In the Maximization phase, all parameters in the CPT are recalculated and this requires calculation of parent counts. This phase is directly proportional to number of parameters. The benefit of using MapReduce depends not only on the size of the input data (as is well known) but also on the size and structure of the network.

Recent studies in BN has focused on local-to-global learning, where the graph structure is learned via one local subgraph at a time. [11] focuses on parallel learning of BN structure by considering multiple learning agents simultaneously. In this setting, each local agent learns one local subgraph at a time. It is observed that parallel learning reduces number of subgraphs required for structure learning. This is done by storing previously queries results and by sharing results between agents. A local learning algorithm focus on a specific target variable and iteratively query other variables to learn the local structure around the target, either its Parent-Child set, Markov Blanket set, or both. Two inference rules query subset and superset inference, and a new parallel Bayesian network structure learning algorithm is proposed.

Work has been done by taking multiple algorithms and running them in parallel on MapReduce and Spark to test for their adaptability [13]. The approach followed is based on a general framework for learning these probabilistic models from large scale and high dimensional data, the latter being a problem with less support in the literature. The difference between MapReduce and Sparks is that while MapReduce relies on hard drives to give intermediate data between every operation, Spark focuses on much more rapid main memory to maintain its data structures.

K2 algorithm is another score and search-based algorithm that has been used for finding Bayesian Network. [14] proposed a new approach for learning of Bayesian Network for Big Data using K2 algorithm. A modification of the KDD process in preparation and pre-processing steps is proposed through the insertion of another stage, in order to optimize the search process in frequency of the data analyzed.

Besides all the improvements in algorithms and parallelism techniques, we can also find some practical examples in literature where these techniques have been being implemented. [9] is an example of Bayesian Network in Natural Language Processing Domain. A technique has been proposed for subjective detection where factual or neutral content is detected using an extension of the Extreme Learning Machine (ELM) paradigm to a novel framework that exploits the features of both Bayesian networks and fuzzy recurrent neural networks to perform subjectivity detection; different to a traditional Laplace approximation technique.

Another example of real-life problem that has been tackled using Bayesian Network is about the ecosystem change in Baltic Sea food web [10]. To find the changes over very small course of time, different series of Dynamic Bayesian Networks with different hidden variables corresponding to different variable structure are fitted. Identifying such changes is a major challenge, as the natural variation in various observed parameters is often high, making it difficult to separate actual data from noise. Hidden variables are variables in the model with no data, linked to the observed variables. They are there to observe the dynamically changing variables that are most likely to effect the BN structure, that is if a value at node changes, or if any change is observed over the period of time, the structure can update itself.

2.3 Hybrid based learning

Hybrid learning comprises of both constraint based and score and search-based approaches. Constraint based approach is used to find conditional independence relation between parent, child and neighboring nodes. Then based on these findings, score and search-based techniques are applied to create a DAG. This resultant DAG is created based on informed decision making as constraint-based algorithm has presented with statistical analysis between nodes. There has not a lot of work done regarding hybrid algorithms and approaches and we can find very few things about it in literature. In [3] a hybrid approach is used for learning of Bayesian network. Relationships between dependent and independent variables is calculated and Conditional Probability Tables are used to evaluate the strength between the relationships of each variable and is a part of parameter learning. MapReduce approach is followed to evaluate the structure and count the probabilities or variables. Once the probabilities are calculated, these are reused in the thickening step for calculating Conditional Mutual Information. Finally, Hill Climbing Algorithm is used to find the optimum structure of Bayesian Network. As the results of probability counting and reduces one pass of MapReduce are reused, this significantly decreases the amount of time required in overall process. Time consumption of the proposed algorithm is near to that of Constraint Based approach and Accuracy or Correctness is near to the that of Search and Score Based approach. Max-Min Hill Climbing (MMHC) is one of the hybrid algorithms that are used for learning of Bayesian Network. [6] takes MMHC and combines it with MapReduce to decrease the time cost of learning a Bayesian network structure. MMHC can be divided into two phases. Phase 1 identifies the skeleton of Bayesian network by conditional independency tests. Phase 2 is searching part where a greedy hill-climbing search performs three different operations, to get the Bayesian network structure; namely add arc, delete arc and reverse arc. For the results boosting method is used that gives the best output after comparing results of iterations.

Chapter 3

Research Problem and Proposed Solution

3.1 Research Problem

Bayesian networks are one of the most useful in creating a model to represent things are done in nature. They provide a good trade off between efficiency and processing power. As Bayesian Network structure learning is an NP-Hard problem in nature, greedy search approach is preferred to get good results in less time.

Max-Min Hill Climbing algorithm is an algorithm that is hybrid is nature; which means that it uses both constraint based and score-and-search based approaches, to get better time performance as that of constraint based approach and good results similar to score-and-search based approach. But as the data size increase and reaches into millions, processing it becomes a difficult and a slow task. Even modern algorithms can't solve the issue. For this we need to incorporate some techniques in order to overcome this issue.

3.2 Proposed Solution

In order to tackle difficult and slow processing of big data, we propose the introduction of Hadoop MapReduce with is a framework for distributed processing of large data also called as big data. It is not possible to convert all the algorithm into distributed processing based solution but we can convert parts of it into distributed and parallel processing framework. Details of implementation will be discussed in next chapter.

3.3 Proposed Methodology

To achieve parallelism, there are many techniques available but when it comes to solving big data problem, there are not many solutions that effectively give results. In addition to that, methodology adopted also plays an important role in overall effectiveness and efficiency of the designed system.

3.3.1 Max-Min Hill Climbing Working

Our proposed solution comprises of two parts, the algorithm and mapreduce framework for achieving parallelism and for dealing with big data problem. The first part, that is MMHC algorithm is described in this section

An graphical representation of the selected algorithm that is MMHC is shown above. This is a high level diagram that shows the important components of the overall algorithm. Detailed working will be explained in the

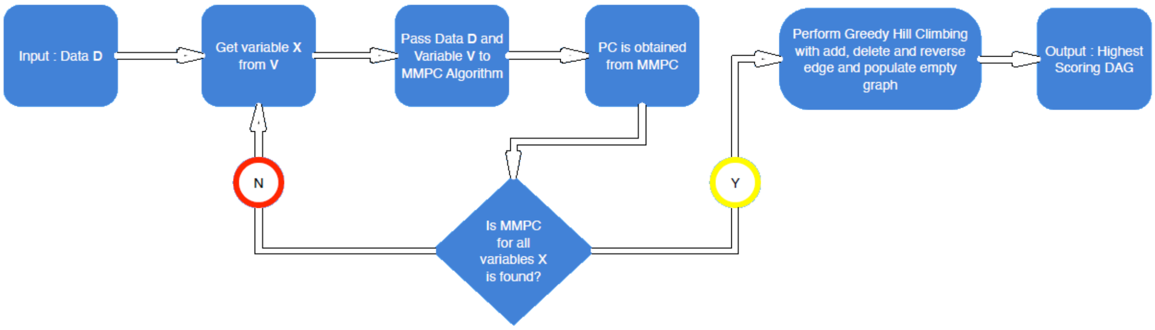
Input : Data D
Get variable X from V
Pass Data D and Variable V to MMPC Algorithm
PC is obtained from MMPC
Perform Greedy Hill Climbing with add, delete and reverse edge and populate empty graph
Output : Highest Scoring DAG
N
Y
Is MMPC for all variables X is found?

Figure 3.1: Max-Min Hill Climbing

next chapter.

As shown in the above figure, the selected algorithm consists of two parts, Max-Min Parent and Children (MMPC) and Greedy Hill Climbing (HC) algorithms. Breakdown of both the algorithms is shown in the figure below.

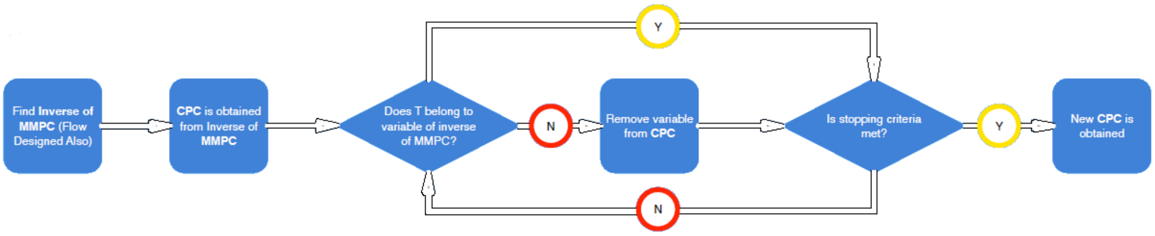
Y
Find Inverse of MMPC (Flow Designed Also)
CPC is obtained from Inverse of MMPC
Does T belong to variable of inverse of MMPC?
N
Remove variable from CPC
Is stopping criteria met?
Y
New CPC is obtained
N

Figure 3.2: Max-Min Parents and Children

The above figure is a graphical representation of all the steps involved in MMPC algorithm. It can be seen that for getting the candidate parents and children set, we need to run the inverse of MMPC algorithm. For given set, we try and find the parent and children set and if the target variable T belongs in the CPC set, we keep them but if it does not, we remove the entry from CPC. Comparison with the target variable is done until we exhaust the CPC set or if we have reached the stopping criteria. The result is a PC set on which greedy hill climbing algorithm is executed.

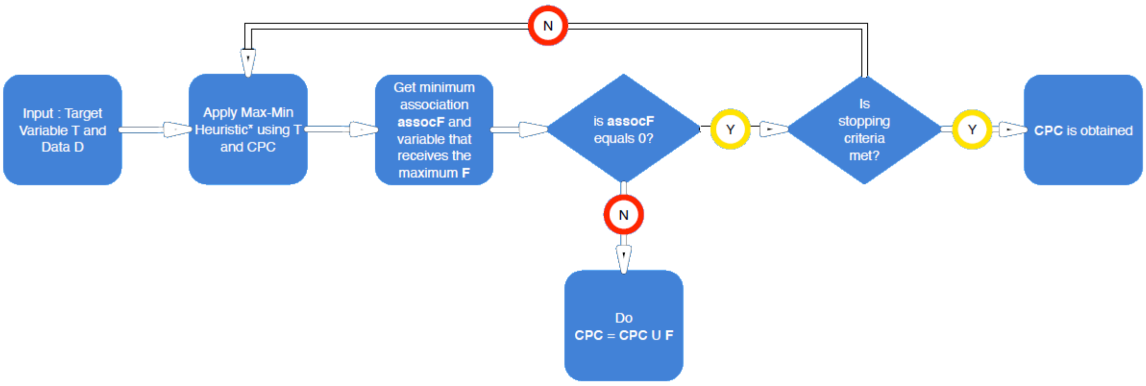
N
Input : Target Variable T and Data D
Apply Max-Min Heuristic* using T and CPC
Get minimum association assocF and variable that receives the maximum F
is assocF equals 0?
Y
Is stopping criteria met?
Y
CPC is obtained
N
Do CPC = CPC U F

Figure 3.3: Inverse of Max-Min Parents and Children - Phase Forward

For finding the inverse of the MMPC algorithm, there are two steps involved, that are phase forward and phase backward. Figure 3.3 shows the first part of the algorithm that is phase forward. Target variable and input dataset is passed to the phase forward and Max-Min heuristics is applied. Result of Max-Min heuristics is minimum association. If association value is zero, we see if the stopping criteria is met, otherwise we add the resultant variable to the CPC set. The second part of finding the inverse of MMPC is phase backward where the input is the CPC set obtained from phase forward step. The first step is to check if the node of CPC is independent of the target variable. If the independence criteria is met, the node is removed from the CPC set and we check if the stopping criteria is met. If the CPC node is not independent of the target variable, that node is kept in CPC set. We contine with this process until stopping criteria is met. Figure 3.4 shows the graphical representation of the process

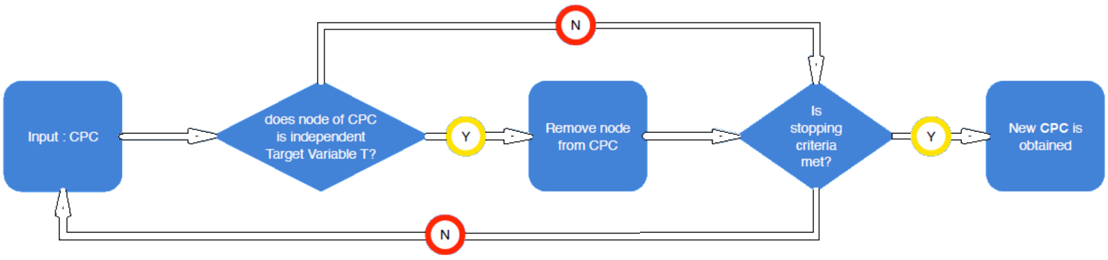
Input : CPC
does node of CPC is independent Target Variable T?
Y
Remove node from CPC
Is stopping criteria met?
Y
New CPC is obtained
N
N

Figure 3.4: Inverse of Max-Min Parents and Children - Phase Backward

Once phase backward is ended, a new CPC set is obtained and second part of MMHC algorithm, that is Greedy Hill Climbing algorithm is executed

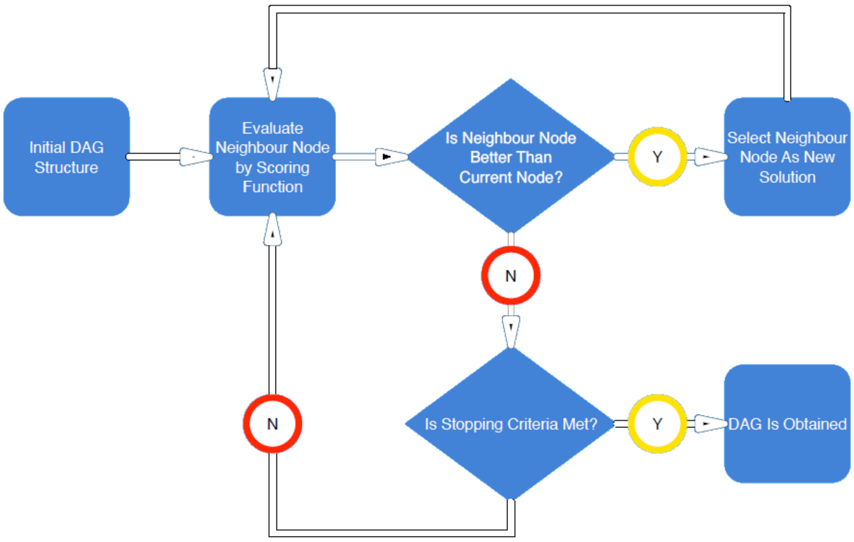
Initial DAG Structure
Evaluate Neighbour Node by Scoring Function
Is Neighbour Node Better Than Current Node?
Y
Select Neighbour Node As New Solution
N
Is Stopping Criteria Met?
Y
DAG Is Obtained
N

Figure 3.5: Greedy Hill Climbing Algorithm

In Greedy Hill Climbing algorithm in MMHC, our goal is to find the resultant Directed Acyclic Graph (DAG). We pass the obtained PC set to the scoring function along with the dataset and we evaluate using any scoring function, if the neighbour node has a better score then the current node in the PC set. If the score is higher then we have a new edge. Similarly, based on the scoring function, we remove or reverse an edge for the provided PC set. Once all the possibilities are checked and PC set is exhausted, we are left with the final DAG which is the output of the MMHC. Figure 3.5 shows a graphical representation of the working of Hill Climbing algorithm.

3.3.2 MapReduce

The second part of out proposed solution is the MapReduce. MapReduce is a framework, that is used for handling large amount of data and process them in chunks, in a parallel manner.

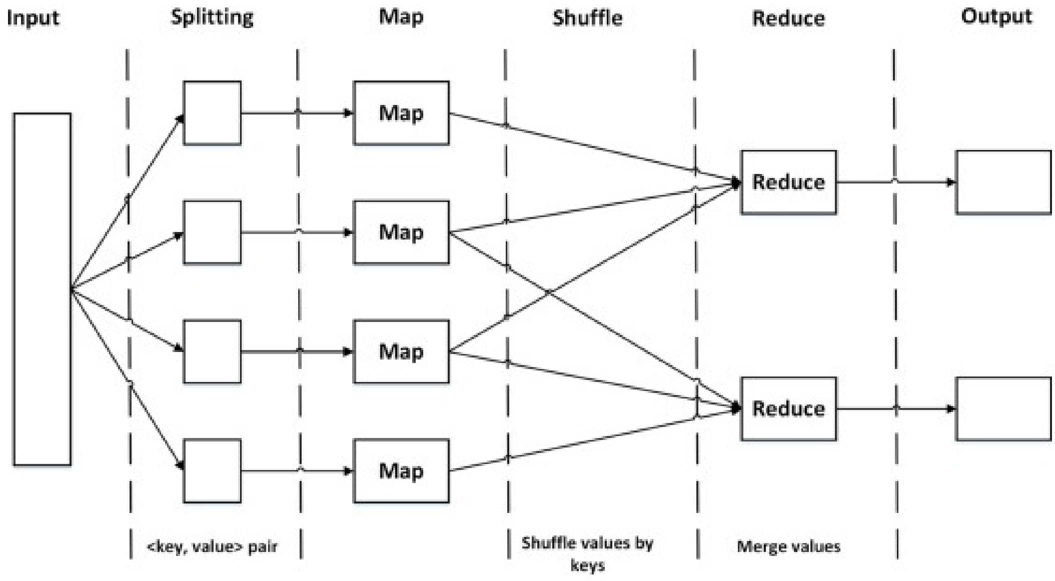

Figure 3.6: MapReduce Flow Diagram

Figure 3.6 shows the flow diagram of how a map reduce works. After the dataset is given as input, the first step is splitting of data into smaller and equal chunks. Once divided, they are mapped as key value pairs. Process of splitting and mapping of data is collectively called as a Mapper Function. Once all the data is mapped in mapper function on key value basis, we pass the data to the Reducer function. Reducer function has two parts; shuffling of the key value pair and reducing the results of similar key value pair together and giving the output.

In our proposed solution, we have adopted the methodology of combining the MMHC algorithm with hadoop mapreduce for processing of large scale data of big data in parallel, in order to reduce the overall execution time needed in sequential processing and also achieving similar accuracy as the traditional approach. Implementation of the algorithm using mapreduce is explained in the next chapter where we discuss how algorithm is modified and how mapper and reduce functions work in order to get the desired output.

Chapter 4

Implementation and Analysis

As discussed earlier, MMHC is a hybrid algorithm in nature. It is a combination of Max-Min Parents and Children, which is a constraint based algorithm and Hill Climbing, which is a score-and-search based algorithm. Both of these algorithms work to create a Bayesian Network or a DAG for a given dataset. A bayesian. network is created as a pair (G,P) where G is a Directed Acyclic Graph (DAG) and P is conditional probability for every given node. DAG consists of G=(V,E), where V represents the given variables from node set and E represents the directed edges between variables of the given note set.

4.1 How algorithm works

As discussed, MMHC comprises of MMPC and HC and is executed in the order as mentioned. MMPC is a constraint based algorithm that uses conditional independency test to determine the relation or dependence between parent and children variables, provided in the given dataset. A contingency table is created to find the conditional independence between variables. Values of the contingency table are calculated using p-value that is based on G-test.

G-test is likelihood-ratio or maximum likelihood statistical significance test that can be calculated using the following formula

$$G = 2\sum_{i} .O_i . \ln\left(\frac{O_i}{E_i}\right)$$

Where $O_i \geq 0$ is the observed count and $E_i > 0$ is the count the is expected to be under the null hypothesis.

Once p-value is obtained, forward update is performed. Based on the p-value, parent and children set is updated. These updated parent and children set are then processed using backward pass where independency test is again performed and whole process of finding p-value and parent and children set based on updated p-value is repeated to get final parent and children set.Once all the parent and children set are obtained, we check for symmetry and pass the PC set for greedy hill climbing phase.

Second phase of MMHC is Greedy Hill Climbing. It searches in the structure space of Bayesian Networks.As Hill Climbing belongs to the family of score and search based algorithms, firstly a scoring function needs to be defined in-order to find the strongest relation between parents and children. In our implementation, we tried two different scoring functions, that are BDeu and BIC and compared the results from both of them. Firstly we find the score of the given PC set using BDeu, which is one of the more widely used scoring approaches. BDeu scoring function is represented as follows

$$BDeu(B,T) = \log(P(B)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma(\frac{N'}{q_i})}{\Gamma(N_ij + \frac{N'}{q_i})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_ijk + \frac{N'}{r_iq_i})}{\Gamma(\frac{N^i}{r_iq_i})}\right)\right)$$

Here P(B) represents prior probability of bayesian structure, n represents number of variables. The other scoring function used for finding score between the given set is BIC. It falls under the category of Information-theoretic scoring functions, are is represented as follows

$$BIC = k\ln(n) - 2\ln(\hat{L})$$

Here $\hat{L}$ represents the maximum likelihood function value, n represents the observations or the sample size and k represents model estimation of number of parameters. Once the scoring functions finds the score, greedy hill climbing starts to add, delete or reverse the edges and whatever leads to the highest score is added to a graph and the search continues. One of the main difference between standard greedy hill-climbing and the one used here is that the addition function is only performed on the edge if it was first discovered by MMPC algorithm. Algorithm terminates when addition, deletion or edge reversal doesn't increase the overall score. Structure having the best score is returned.

4.2 MapReduce

A MapReduce is a framework, designed for processing large amount of data in parallel by dividing into smaller parts. As the name suggests, MapReduce works using two main steps: Map and Reduce. Map function, which is a user written function, takes dataset as input and performs splitting of data and creating key value pairs. As Map function takes one value record as input, we can parallelize it completely.

These key value pairs are then passed to a user written reduce function. Reducer shuffles the record and is passed to reducer where same keys are grouped together and output is given.

4.3 Algorithm Implementation with Map Reduce

In the section, key parts of Max-Min Hill Climbing algorithm using MapReduce will be discussed. As the algorithm comprises of two parts that is Max-Min Parent and Children algorithm and Hill Climbing algorithm, we have make use of MapReduce for some parts. We divide overall algorithm into steps

Step 1 and 2 of the algorithm comprises of Independence tests for MMPC for forward and backward phases. In forward phase, we make conditional independence test parallel and pass the data to the MR framework. As discussed earlier, in Mapper phase, two steps are involved namely splitting and mapping. Data is split into smaller chunks and then conditional independence test is performed on the PC set and candidates set. p_value is calculated against every candidate variable. Map function then assigns each calculated value for against a key, creating a key-value pair. This set of key-value pair is the passed to the Reducer where shuffling is preformed and all the similar keys are groups together and passed to the reduce function. Reduce function then sums all the values for similar keys and gives independence test result. Once p_value is calculated from independence test, it is passed to update forward function where we get updated PC and Candidates sets.

Once we get result from MMPC forward phase, we pass the updated PC and Candidates sets to MMPC backward phase. We again use the same MR implementation for independence test as MMHC forward phase. In backward phase, we transfer the variable in PC set to Candidate set except the last one and perform independence test and pass the results to the update backward function to get updated PC and Candidates set.

Once we obtain the required outputs, we check PC set for symmetry to get better results. Once we have results, we pass the dataset, PC set and scoring function parameter to hill climbing algorithm that finds best possible DAG. Hill climbing algorithm then adds, removes or reverses the edges based on scoring function. The resultant is a graph which can then be drawn to get a pictorial representation.

High-level algorithms are given to give a more clear picture.

Algorithm 4.1 Main MMHC Process

```
PC = {} //Empty Parent Children set
D : Data set
Output : Bayesian Network Structure
Step 1
Pass dataset, empty PC set, target variables and candidate set to MMPC Forward Pass
for each d in D:
    IT_MR() //Perform Independence Test using MapReduce
    p_value = Update_ForwardPass() // p_value is obtained as a result

    // PC and Candidate set obtained as a result

Step 2
Pass dataset D, updated PC and candidate set, and target variables to MMPC Backward Pass
for each d in D:
    IT_MR() // Independence Test using MR with updated values
    p_value = Update_BackwardPass() Updated p_value is obtained as a result
    // Updated PC and Candidates set are obtained as a result

//Update symmetry of PC set
Step 3
//Pass dataset, PC set and scoring function to Hill Climbing Algorithm
DAG = HC(dataset, PC, score_function) // Directed A-Cyclic Graph is //returned as an output
```

Pseudo code for Conditional Independence Test using MapReduce is given below

Algorithm 4.2 IT_MR() :Conditional Independence Test using MapReduce

```
Map(Dataset D, PC_Set P, Candidate_Set C)
for each d in D:
    p_value = Find_IndependenceTest() //Find Conditional
    //Independence Test on the PC and Candidate Set
    return (Key_Value Pair) //p_value for a given PC and
    //Candidate Set
end for

Reduce (Key_Value Pair Set From Mapper)

for each subset value in :
    total = Sum(value list)
    Return (total)
end for
```

As MapReduce is used for calculating Independence Test, this reduces the overall execution time with relatively similar accuracy as that of sequential execution of MMHC.

Chapter 5

Results and Discussion

Max-Min Hill Climbing itself is a hybrid algorithm in nature and comprises of constraint based and score and search based approach. This means that the average running time would be better then a traditional score and search based algorithm but when it comes to accuracy, it would have better results as compared to constraint based algorithm.

With the introduction of MapReduce, and by making conditional independence test parallel the overall running time of the hybrid bayesian network very less as compared to the sequential execution, and by keeping the overall accuracy similar. We ran the data-sets on both sequential and parallel algorithms and measured execution times for both parts, that is constraint based score and search based parts of MMHC, for both the approaches. Data-sets we ran on our implementation were of alarm, which is a bayesian network have patient monitoring system data and hailfinder, which is a bayesian network for the forecasting of sever hail storm in summer. Variables in both the data-sets are 37 and 56 respectively. The given data-set sizes are of 10,000 and 5,000 records respectively, we generated 1, 5 and 10 million rows as sample dataset for our experiment. The algorithm was run multiple times for data sets and the total execution time is the average over all the executions. Here we have shown two different graphs for two different data-sets and it is clear from the graphs that execution time of the MapReduce approach is much less then that of non MapReduce execution.

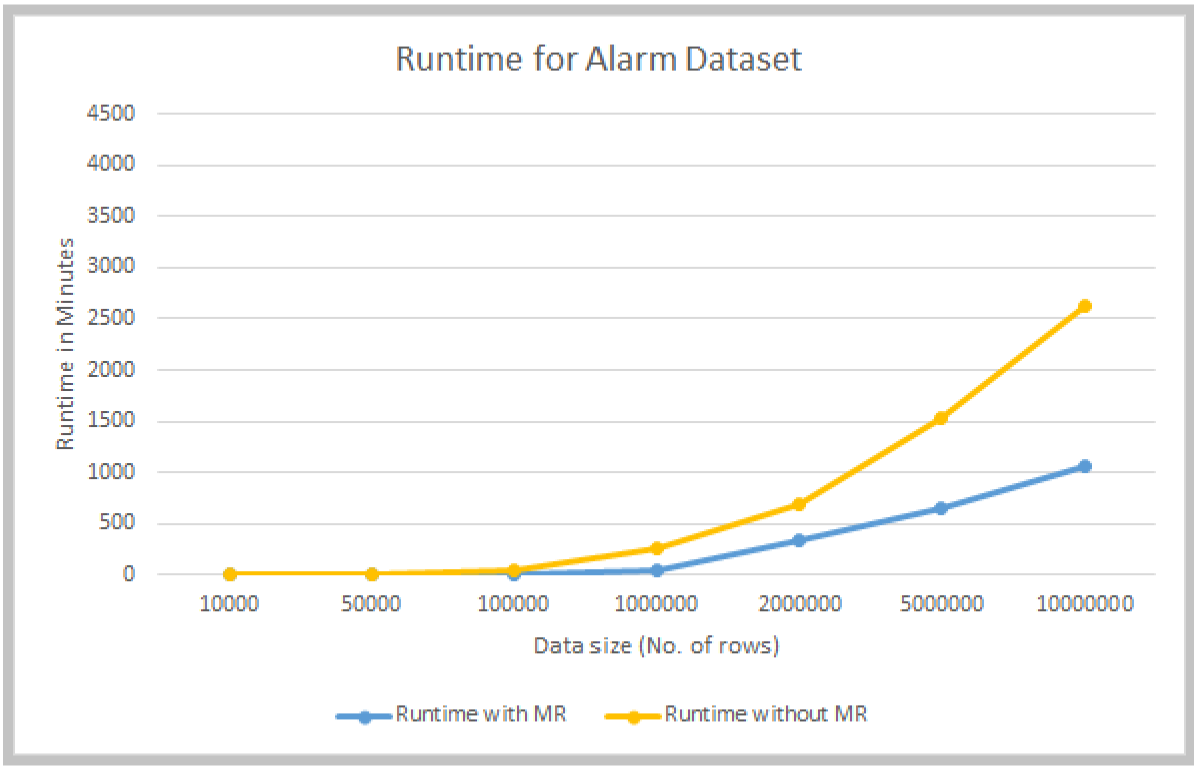

Figure 5.1: Execution Time for Alarm Dataset

From the graph, we can see that as the data size increases, the running time for sequential approach increases drastically as compared to MapReduce based approach. Similarly, if we see the graph for hailfinder data set, we can see similar trends.

One thing we observe between the two graphs is that there is a lot of difference in execution time for same size of data. This is because of the difference in the size of number of variables. Alarm dataset has 37 variables as hailfinder has 56 variables. This is one of the major contributing factor in difference of execution times. Results for accuracy for two different approaches are not

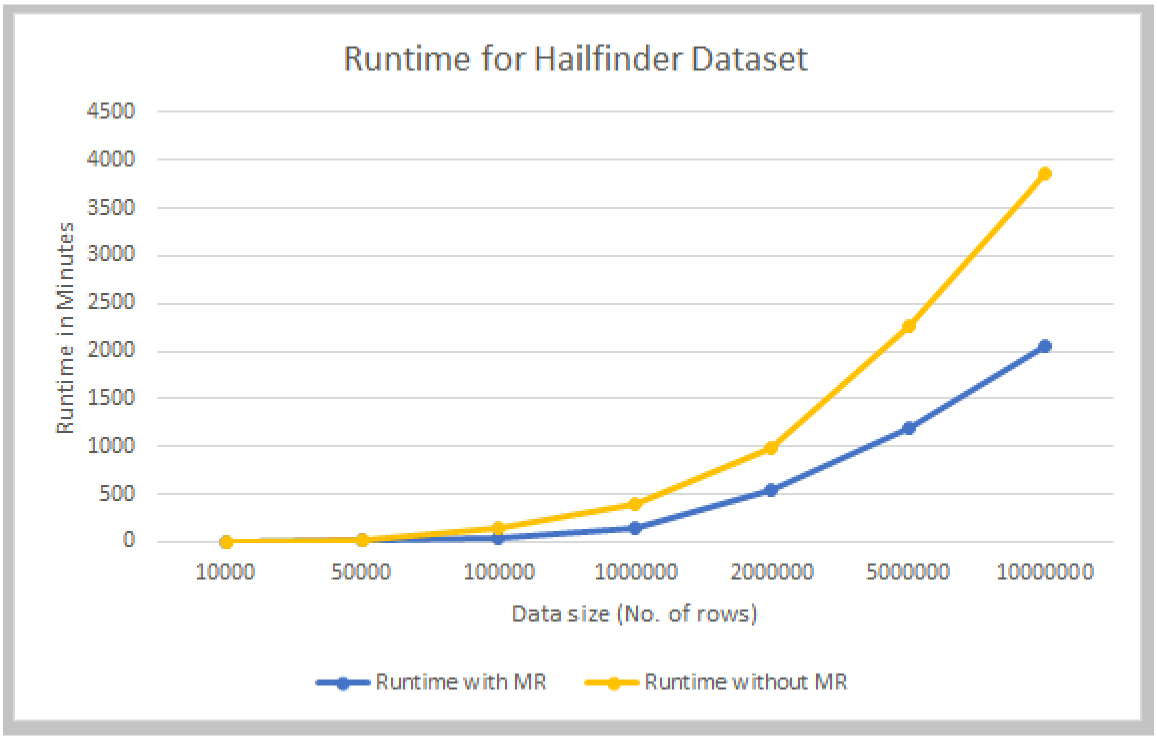

Figure 5.2: Execution Time for Hailfinder Dataset

been added here as there is not a very visible difference. This is due to the fact that the we have used same MMHC algorithm and have made it parallel to achieve speedup and from the graphs, it is visible that our execution times have been reduced.

It was interesting to note that the execution time for dataset having smaller sizes like 5000 to 10000 had better execution timings when ran sequentially then when running in parallel. It was found out that as the computers are getting more powerful, they are better in solving small size data problems when executed in sequential manner rather then dividing the dataset into smaller chunks, performing the required task and group them together to get the output.

This experiment was only performed for Max-Min Hill Climbing as there is enough evidence in the literature that a hybrid algorithm performs better then score-and-search based algorithms when it comes to accuracy and shows significant improvement in execution time as compared to constraint based algorithm. So, we believe that this will hold true even for parallel based approach.

For the experiment, Hadoop 3.7 was installed on three systems, running Windows 10 Pro. For the namenode, the system used was was Intel core i7-8565U processor @1.99GHz with 8GB RAM and for two datanodes Ryzen 7 5800x with 64GB 3600 MHz CL16 RAM and RTX 3080 graphics card. As there are only two datanodes, and as the namenode system is not a powerful one, this resulted in higher execution times. By having a more powerful namenode a high number of datanodes, the execution time can be reduced drastically as the data would be divided even more, hence achieving greater level of parallelism.

Chapter 6

Conclusion and Future Work

In this paper, we have established that using parallelism techniques such as MapReduce with a hybrid bayesian network structure learning algorithm, we can get better execution timings then any sequential approaches. Experimental results were carried out using sample data-sets and the results show that we can easily incorporate parallelism technique for real life data and can achieve similar accuracy with better speedup as compared to traditional sequential approaches. Also, we concluded that with the addition of more datanodes, the overall execution timings can be further reduced, especially for data having high number of variables.

6.1 Future Work

In future, we can create more hybrid algorithms by combining constraint based and score-and-search based algorithms, besides algorithms like MMHC that are already hybrid in construction and can see which approach gives better accuracy as compared to MMHC and lower the overall execution time.

Bibliography

[1] J. A. Gámez, J. L. Mateo, and J. M. Puerta, "A fast hill-climbing algorithm for bayesian networks structure learning," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty.* Springer, 2007, pp. 585–597.

[2] K. Yue, Q. Fang, X. Wang, J. Li, and W. Liu, "A parallel and incremental approach for data-intensive learning of bayesian networks," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2890–2904, 2015.

[3] S. Li and B. Wang, "A method for hybrid bayesian network structure learning from massive data using mapreduce," in *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids).* IEEE, 2017, pp. 272–276.

[4] Q. Fang, K. Yue, X. Fu, H. Wu, and W. Liu, "A mapreduce-based method for learning bayesian network from massive data," in *Asia-Pacific Web Conference.* Springer, 2013, pp. 697–708.

[5] A. L. Madsen, F. Jensen, A. Salmerón, H. Langseth, and T. D. Nielsen, "A parallel algorithm for bayesian network structure learning from large data sets," *Knowledge-Based Systems*, vol. 117, pp. 46–55, 2017.

[6] J. Hu, G. Wu, P. Sun, and Q. Xiong, "A parallel bayesian network learning algorithm for classification," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2016, pp. 259–263.

[7] A. Basak, I. Brinster, X. Ma, and O. J. Mengshoel, "Accelerating bayesian network parameter learning using hadoop and mapreduce," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, 2012, pp. 101–108.

[8] M. Scanagatta, G. Corani, C. P. De Campos, and M. Zaffalon, "Approximate structure learning for large bayesian networks," *Machine Learning*, vol. 107, no. 8, pp. 1209–1227, 2018.

[9] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria, "Bayesian network based extreme learning machine for subjectivity detection," *Journal of The Franklin Institute*, vol. 355, no. 4, pp. 1780–1797, 2018.

[10] L. Uusitalo, M. T. Tomczak, B. Müller-Karulis, I. Putnis, N. Trifonova, and A. Tucker, "Hidden variables in a dynamic bayesian network identify ecosystem level change," *Ecological Informatics*, vol. 45, pp. 9–15, 2018.

[11] T. Gao and D. Wei, "Parallel bayesian network structure learning," in *International Conference on Machine Learning.* PMLR, 2018, pp. 1685–1694.

[12] M. Scutari, C. Vitolo, and A. Tucker, "Learning bayesian networks from big data with greedy search: computational complexity and efficient implementation," *Statistics and Computing*, vol. 29, no. 5, pp. 1095–1108, 2019.

[13] J. Arias, J. A. Gamez, and J. M. Puerta, "Learning distributed discrete bayesian network classifiers under mapreduce with apache spark," *Knowledge-Based Systems*, vol. 117, pp. 16–26, 2017.

[14] A. S. De França, J. G. R. d. O. Lima, A. F. J. Junior, and Á. L. De Santana, "Learning the bayesian structure in bigdata using the k2 algorithm with mapreduce," in *Proceedings of World Congress on Systems Engineering and Information Technology*, vol. 1, 2013, pp. 15–19.

[15] M. Scutari, C. E. Graafland, and J. M. Gutiérrez, "Who learns better bayesian network structures: Constraint-based, score-based or hybrid algorithms?" in *International Conference on Probabilistic Graphical Models.* PMLR, 2018, pp. 416–427.

[16] M. Scanagatta, A. Salmerón, and F. Stella, "A survey on bayesian network structure learning from data," *Progress in Artificial Intelligence*, vol. 8, no. 4, pp. 425–439, 2019.

[17] M. Scutari, "Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn r package," *arXiv preprint arXiv:1406.7648*, 2014.

[18] M. O. Shafiq, Y. Yang, and M. Fekri, "A survey and recommendations for distributed, parallel, single pass, incremental bayesian classification based on mapreduce for big data," in *2017 IEEE 19th International Conference on High Performance Computing and Communications Workshops (HPCCWS).* IEEE, 2017, pp. 42–49.

[19] T. Rahier, S. Marié, S. Girard, and F. Forbes, "Fast bayesian network structure learning using quasi-determinism screening," in *JFRB 2018-9èmes Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes*, 2018, pp. 14–24.

[20] A. Basak, I. Brinster, and O. J. Mengshoel, "Mapreduce for bayesian network parameter learning using the em algorithm," 2012.

[21] 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.

[22] K. Liu, Y. Cui, J. Ren, and P. Li, "An improved particle swarm optimization algorithm for bayesian network structure learning via local information constraint," *IEEE Access*, vol. 9, pp. 40 963–40 971, 2021.

[23] S. Behjati and H. Beigy, "Improved k2 algorithm for bayesian network structure learning," *Engineering Applications of Artificial Intelligence*, vol. 91, p. 103617, 2020.