International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Parallelizing Coherent Rule Mining Algorithm on CUDA

Aditya A. Davale[1], Shailendra W. Shende[2]

[1, 2] Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India.

Abstract: ***In data mining, association rules are discovered from the domain when the minimum support threshold value is given by domain expert. Minimum support threshold affects the number of rules generated and the quality of rules. So In this work, a framework is proposed to discover domain knowledge in terms of coherent rules. This rules are discovered from properties of propositional logic, and does not require minimum support threshold.The execution of association rule mining algorithm involves processing of huge data.Handling such amount of data having millions of rows is very challenging. High-end computers and server-side machines are used to process such algorithms for enormous amount of data but are very expensive and accessible to only a few. In comparison graphics processing units (GPUs) have much high computation power and are also less expensive. So, GPU acts as the high performance co-processors. So, the implementation of association rule mining algorithm on GPU will provides the high performance computing and increase in the speedup.***

Keywords: Data mining, GPU, CUDA, parallel processing.

1. Introduction

The use of association rule mining technique is to describe the associations among items in a database. These associations represent the domain knowledge encapsulated in databases. Identifying domain knowledge is important because these knowledge rules usually are known only by the domain experts over years of experience. Thus, association rule mining is useful to identify domain knowledge hidden in large volume of data efficiently [1].

The discovery of association rules is typically based on the support and confidence framework where a minimum support (min sup) [2] must be supplied to start the discovery process. Apriori is a representational algorithm based on this framework and many other algorithms are a priori-like [2]. Without this threshold specified, typically, no association rules can be discovered because the procedure to discover the rules will quickly exhaust the available resources.

In this work, a framework is proposed to address the above issues by removing the need for a minimum support threshold. Association rules are discovered based on propositional logic. The principle of the approach considers that an association rule should only be reported when there is enough logical evidence in the data. To do this, the presence and absence of items during the mining is considered. An association such as beer → nappies will only be reported if we can also find that there are fewer occurrences of beer → nappies and ¬ beer → ¬ nappies but more of beer → nappies. This approach will ensure that when a rule such as beer → nappies is reported, it indeed has the strongest statistical value in the data as comparison was made on both presence and absence of items during the mining process [1].

Data mining algorithms involve processing of huge data for association and many other purposes. In practice, handling such amount of data having millions of rows is very challenging. High-end computers and server-side Computers are used to process such algorithms for enormous amount of data but are very expensive and accessible to only a few. In comparison graphics processing units (GPUs) have much high computational power and are also less expensive [5]. They are not only part of Server Computers but are also available in almost every Desktop and Laptop that establishes a default computing environment for parallel processing. Presently multi core CPUs are available in two and four core packages at very low cost. Similarly, current GPU implementation ranges from 768 to 12288 concurrently executing threads which if properly utilized can speed up the application at exponential levels. GPUs have great potential as high-performance co-processors. Researchers have focused on efficient and highly scalable implementations of different data mining algorithms. Among these, a major approach taken is the development of parallel and distributed versions of algorithm [5].

So, the implementation of Association Rule mining algorithm using GPU CUDA based architecture and its performance analysis to demonstrate which implementation is best i.e., CPU vs. GPU[5].

2. Algorithm

Coherent rules are found by using properties of positive and negative rules on the condition set (positive rule) > set (negative rule) at preselected consequence item set. The algorithm [1] is given in the Fig.1

2.1 Steps of Serial Algorithm

1. Assign Coherent Rule set to NULL.
2. Generate the Power set of the given Attributes.
3. Take the pair(X,Y) and find the support (X,Y) and (~X,Y) from powerset.
4. Compare the support of both and generate the rule accordingly and add the generated rule to the Coherent Rule set.
5. Repeat the step 3 and 4 until all the elements in the power set are visited.

Input: D – a database, Y – a consequence item set
Output: CR – a set of coherent rules

[1] $CR \leftarrow \emptyset$
[2] $I \leftarrow$ find a set of unique items from D
[3] Let $A = I - Y$
[4] $Y.count \leftarrow$ total counts of Y in D
[5] $O_{P(A)} \leftarrow$ virtually map the power sets of A to the indices of a binary system
[6] For each i-th element of the power sets of A in order of $O_{|}$,
(i) $X \leftarrow \{P_i : i \in P(A)\}$
(ii) $S(X,Y) \leftarrow XY.count$
(iii) $S(\neg X,Y) \leftarrow Y.count - S(X,Y)$
(iv) if $S(X,Y) > S(\neg X,Y)$,
if equation (2) is met, $CR = CR \cup (X,Y)$
Loop [6] until $i = |P(A)|$
(v) remove all power sets of A having the i-th element
[7] return CR

* For example, given 3 items, the first item set *null* – a member in the power sets of X, item set $X_{\neg\neg\neg}$ is indexed using binary number '0', item set $X_{\neg\neg}$ is indexed using '1,', and item set $X_{\neg}$ is indexed using '10,'.

Figure 1: A simple coherent rule mining algorithm (CH search)

2.2 Features of algorithm

1. Does not require minimum support threshold.
2. Does not need to generate the frequent itemsets.
3. Identifies negative association rules.

2.3 Proposed parallel coherent rule mining algorithm

1. Assign Coherent Rule set to NULL.
2. Generate the Power set of the given Attributes in parallel.
3. Take the pair(X,Y) and find the support (X,Y) and (~X,Y) from for all combinations in powerset in parallel.
4. Compare the support of both and generate the rule accordingly and add the generated rule to the Coherent Rule set in parallel.

2.4 Working Example

Suppose, as a manager of stationary store, you would like to determine which items are frequently purchased together. So, the rules are found that have frequent purchases. The sample database, is shown in Table 1.Now, from the sample database, we know that
No of records =10
No of attributes = 3
So power set of 3 attributes contains 2^3=8 combinations.

Table 1: A sample database

Sr. No	Pen	Pencil	Eraser
1	1	1	0
2	1	0	1
3	1	1	1
4	0	0	1
5	0	0	1
6	1	1	1
7	1	1	0
8	0	1	1
9	1	0	1
10	1	1	0

Let's find the support for pen=1, pencil=1, eraser=0
Positive rule: support (pen=1, pencil=1, eraser=0) →0.3
Negative rule: support (pen=0, pencil=0, eraser=1) →0.2
So here, 0.3 > 0.2 so the positive rule is chosen over negative rule. The same scenario is continued for the remaining combinations in the power set.

3. Experimental Setup

This chapter includes the information about the dataset used for coherent rule mining algorithm, configuring the Computer for implementing parallel code on CUDA cores, Algorithm implementation.

3.1 Details of Dataset

We are using the 'zoo' dataset collected from UCI repository [10].The dataset contains 15 Boolean valued attributes and type attribute which is the class attribute. There are 101 transactions in the dataset. The dataset is multivariate and the attributes are categorical and integer. The dataset contains no missing attribute values and the class distribution is on the type attribute.

3.2 Configuration

The coherent rule mining algorithm is implemented on the Intel core i5 processor with NVIDIA GT 520MX graphics card installed on motherboard. The operating system used is Cent OS 6.5 (64 bit).

3.3 Parallel Algorithm Implementation

Following steps are performed for parallel implementation of coherent rule mining algorithm:
1. Allocate the memory on cpu by using malloc function
2. Allocate the memory on gpu by using the cudamalloc function with passing the appropriate parameters to the function.
3. Read the dataset from file to array on cpu.
4. Generate the powerset on gpu by invoking the kernel function of powerset generation and transferring the dataset on gpu by using the function CudaMemCpy with parameter of CudaMemcpyHostToDevice.This will give the all possible combinations of the attributes. Take the results back on the cpu by CudaMemCpy function with parameter of CudaMemcpyDeviceToHost [8].
5. Find the positive and negative support for all the possible combinations in parallel by invoking the kernel function called positive rule generation and negative rule generation and transferring the dataset and all possible combinations on the gpu by by using the function CudaMemCpy with parameter of CudaMemcpyHostToDevice.This will give the positive and negative support for each combination. Take the results back on the cpu by CudaMemCpy function with parameter of CudaMemcpy HostToDevice [8].
6. Compare the positive and negative support and generate the rule where support is greater.

4. Result and Analysis

In this chapter, the results are taken from two Computers. The results are taken with considering the data transfer time on gpu and without considering data transfer time on gpu. In this chapter "WITHOUT DT" means speedup calculated without considering the data transfer time and "WITH DT" means speedup calculated with considering the data transfer time. The configurations of Computer #1 and Computer #2 are given below.

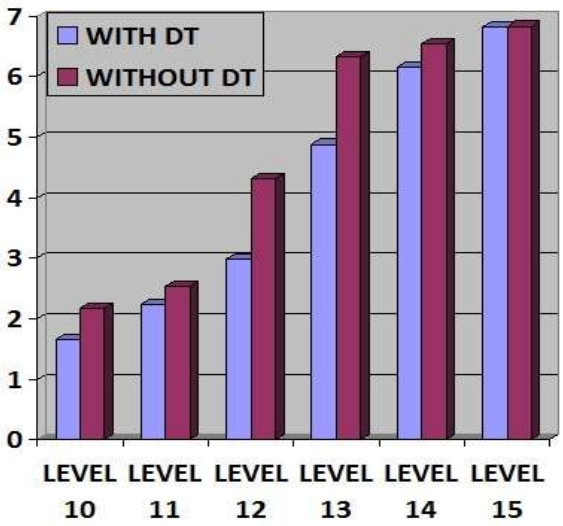

Figure 2: Change in speedup with respect to levels for Computer #1

Computer #1: CPU: Intel core i5 2.5 Ghz, 4GB RAM.GPU: Nvidia GT 520mx, 1GB RAM

The graph shown in figure 2 is the graph of speedup at various levels for Computer #1. The levels on the X-axis indicates that
the no of items in that itemset. Hence level 10 means there are
10 items in that itemset. On the Y-axis there is a speedup factor. The blue bar indicates the gpu speedup with data transfer on gpu and the magenta bar indicates the speedup without data transfer. So here we can see that as the levels are increased the speedup also increases.

Computer #2: CPU: 2.0 Ghz four core intel Nehalem,16GB RAM GPU: Nvidia quadro FX 380LP

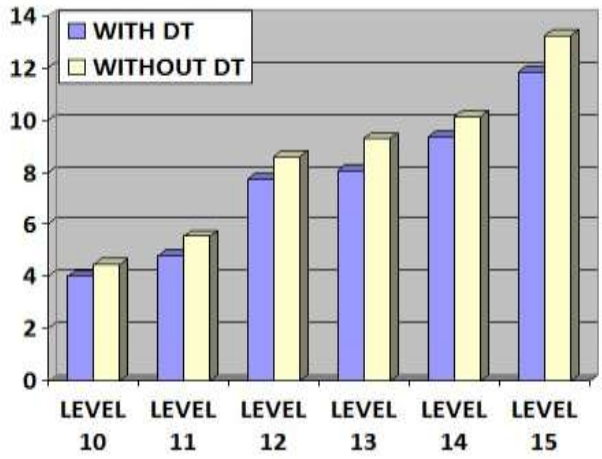

Figure 3: Change in speedup with respect to levels for Computer #2

The graph shown in figure 5.2 is the graph of speedup at various levels for Computer #2. The levels on the X-axis indicates that the number of items in that itemset. On the Y-axis there is a speedup factor. The blue bar indicates the speedup with data transfer. The yellow bar indicates the speedup without data transfer. So here we can see that the speedup achieved without transfer of data is more as compared to speedup achieved with data transfer.

The speedup gain by both the Computers is compared in the graph shown in figure 4.From figure 4, we can say that the speedup achieved on the Computer #2 is better as compared to the Computer #1. From all the above observations it is clear that Parallel algorithm gives better results as compared with serial once. The performance of parallelization is responsible for reduction in time and provides same results.

5. Conclusion and Future Scope

Data mining is a technique for mining hidden knowledge from large number of databases, which is never seen before. Data mining is mostly used for knowledge discovery. Association

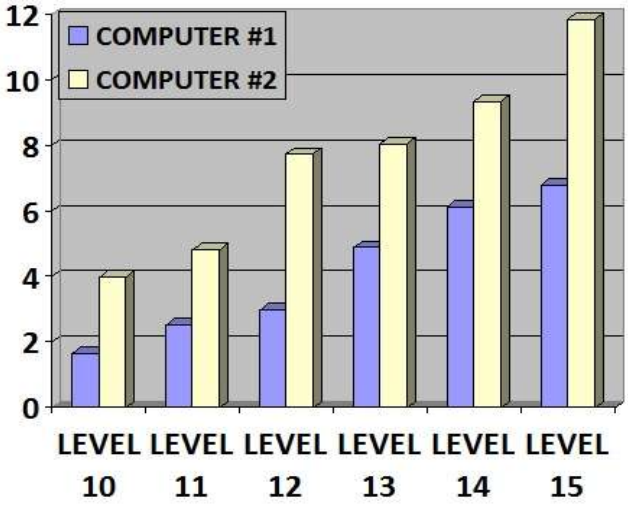

Figure 4: Speedup for both the Computer #1 and Computer #2

rule mining (ARM) is one of the functionality of data mining. Association rule mining concept is used to show relation between items in a set of items. Coherent rule mining algorithm is used for mining itemsets from large amount of dataset. Data mining in a Serial manner can consume time and reduce performance for mining. Parallelism is used to reduce time and increase performance. Parallelizing Coherent rule mining algorithm has improved the performance of this algorithm drastically. In this work, the serial and parallel performance of Coherent rule mining algorithm is compared with respect to time. Results of parallel algorithm are better than serial algorithm with respect to time. Thus, Parallelizing has increased the performance. The speedup achieved by the Computer #2 is almost 11.86 for level 15.

Though, this much speedup is achieved, the algorithm can be run on very high end graphics processors for best speedup. The same algorithm can be implemented in parallel on multi-core processors using OpenMP and compare the results of parallel implementation of this algorithm on CUDA and OpenMp.

References

[1] Alex Tze Hiang Sim, Maria Indrawan, Samar Zutshi, ”Logic-Based Pattern Discovery”. IEEE transaction on knowledge discovery and data engineering, VOL. 22, NO. 6, JUNE 2010
[2] R. Agrawal, T. Imielinski, and A. Swami, ”Mining Association Rules between Sets of Items in Large Databases,” SIGMOD Record,vol. 22, pp. 207-216,1993
[3] C. Longbing, ”Introduction to Domain Driven Data Mining,” Data Mining for Business Applications, L. Cao,P.S. Yu, C. Zhang, and H. Zhang, eds., pp. 3-10, Springer, 2008
[4] B.Liu, W.Hsu, and Y.Ma, ”Mining Association Rules with Multiple Minimum Supports,” Proc. ACM SIGKDD, pp. 337-341, 1999
[5] Syed Hasan Adil,Sadaf Qamar ,“Implementation of Association Rule Mining using CUDA” 2009 International Conference on Emerging Technologies
[6] Jayshree Ghorpade, Jitendra Parande, Madhura Kulkarni, Amit Bawaskar” Gpgpu Processing In Cuda Architecture” Advanced Computing: An International Journal (ACIJ), Vol.3, No.1, January 2012
[7] J. Han, M. Kamber. Data Mining: concepts and techniques, Beijing: China Computer,Press,2006
[8] CUDA C PROGRAMING GUIDE 2014
[9] Website: http://docs.nvidia.com/cuda/cuda-c-programming-guide
[10] CUDA Toolkit Website: https://developer.nvidia.com/cuda-zone
[11] UCI Repository Zoo Dataset Website: https://archive.ics.uci.edu/ml/datasets/Zoo