Data Mining Techniques in Parallel Environment- A Comprehensive Survey

Kinjal S. Shah
M. Tech Student,
Institute of Technology,
Nirma University

Prashant Chauhan
Project Scientist,
BISAG
Gandhinagar, Gujarat

M. B. Potdar, Ph.D.
Project Director,
BISAG
Gandhinagar, Gujarat

ABSTRACT
Data mining is the process of discovering interesting and useful patterns and relationships in large volumes of data. The valuable knowledge can be discovered through the process of data mining for the further use and prediction. We have different data mining techniques like clustering classification and association. Classification is one of the major techniques to discover the patterns in huge amount of data. This technique is widely used in many fields. We have a large volume of data and if we extract the data sequentially then it will take a lot of timing. So if we extract the data parallely, the amount of time taken can be reduced. We can use parallel techniques when there is a large volume of data and we want to extract the data in very few seconds. We can implement this techniques using different approaches like MPI, OPENMP, using CUDA or using Map Reduce approach. Here in this paper we will discuss data mining techniques classification by decision tree induction and k- nearest neighbors using both sequential approach as well as parallel approach.

General Terms
Data mining, Classification

Keywords
Decision tress, KNN, MPI, CUDA, KDD, OPEN MP

1. INTRODUCTION
Organizations need to accumulate a large and continuously growing data in different databases. This data can be of any type, this data may be either transactional database like payroll, sales, accounting etc. or it may be analytical data which is helpful in decision support system. For the purpose of utilizing this data, they must be analysed thoroughly. Many analytical tools are already available in market. We also can retrieve valuable knowledge with the use of data mining techniques that can be useful for further use and prediction. Data mining is considered as a part of Knowledge Discovery in Database process. Main steps of KDD include data accumulation, data cleaning, pre-processing, storing, mining and representing the patterns in a presentable format.

Uses of data mining system:

1) *Business Transactions*: Every transaction in the business needs effective use of data in a reasonable amount of time. So here we can use data mining techniques to make the transactions in very short time [1].

2) Scientific Data: Now days many research is going on. So we need to analyse scientific data. We can capture and store more new data faster and can analyse them.

3) Medical and Personal Data: From government and private sectors, we have very large collection of the patients available. We can recognize the particular disease from this data and we can use them worldwide.

4) Surveillance Video and Pictures: For the purpose of storing video tapes and digitalizing them for the future use and analysis.

5) Text Reports and Memos: Most of the communication today is text based. So we can store them in digital format and retrieve them when they are needed.

6) Satellite Sensing: There are numbers of satellites around the globe and they are continuously sending the data to the surface. By the use of data mining we can make the data publicly available and many scientists can research on them.

7) The World Wide Web Repositories: In World Wide Web, we have all type of formats, contents and descriptions are collected and connected with hyperlinks. Here most of data available worldwide. So we can use data mining techniques over there.

Here, the data set will be an image. So we need to process that image to get useful data for the prediction. Real world digitized geo-spatial data can be stored in two basic forms with an optional temporal component viz. raster and vector. [10]

Raster Data: ESRI (Environmental Systems Research Institute) defines Raster as a spatial data model that defines space as a 2D array of equally sized cells arranged in rows and columns, and composed of single or multiple bands. It describes the information through values stored in pixels. The spatial resolution of a raster image is dependent upon the resolution of the acquisition device such as Optical Sensor, CCD Device or other imaging device and its quality upon the source of data. [10]

Vector Data: The vector data describes information through geometric shapes such as point, line, multiline, polygons and other complex shapes. It is mostly prepared through surveying and digitization of maps manually or through supervised/non supervised automated programs. Pattern Recognition and Image Processing techniques are used to convert raster formats into vector formats whereby vector features like lines and polylines are identified by the tracing program. Recent advances in image processing algorithms helps to convert much of the raster data into vector formats with very high acceptable accuracy.

2. ARCHITECTURE AND PROCESS OF DATA MINING

2.1 Architecture of Data Mining System

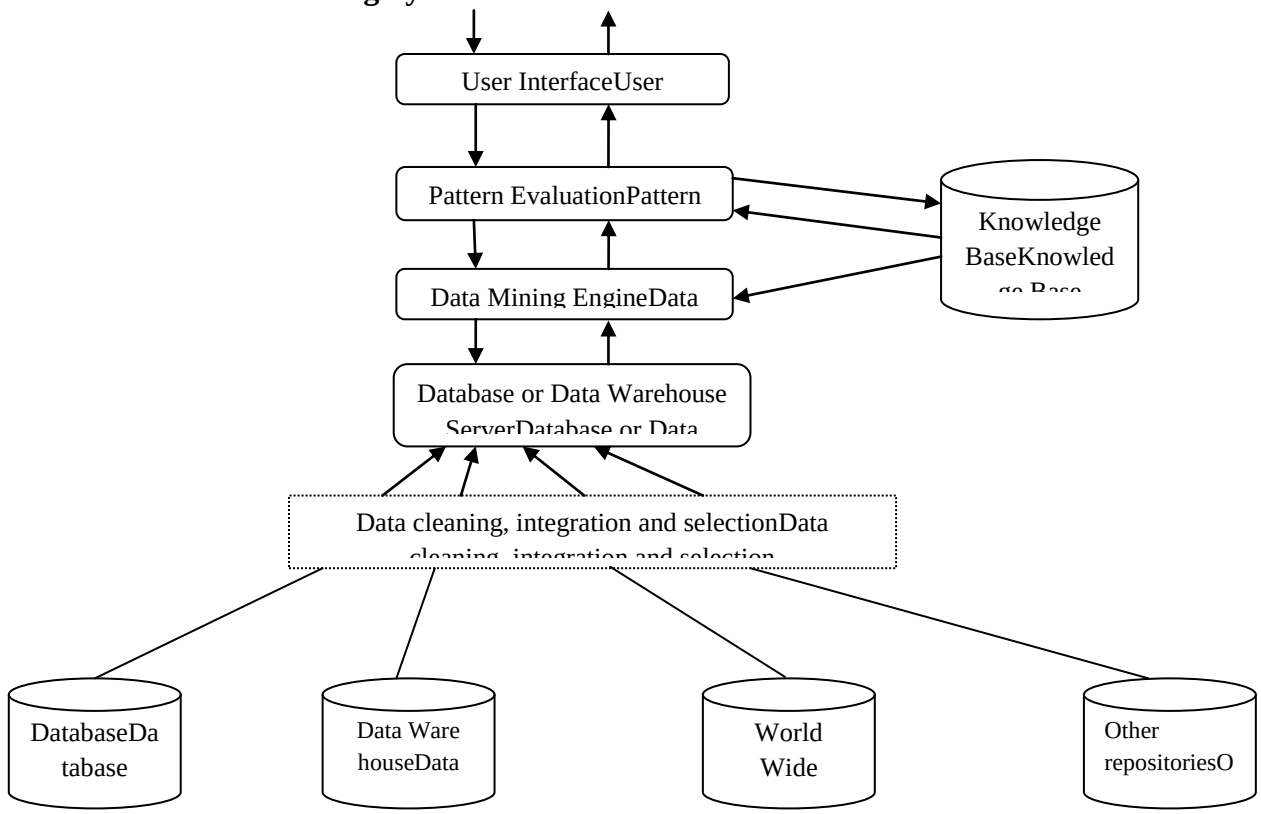

Figure 1: Architecture of data mining system

Architecture of data mining system has following components:

1) Database, data ware house, or other information repositories: This is one or a set of databases, data warehouses, spread sheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

2) Database or data ware house server: The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

3) Knowledge base: This is the domain knowledge that is used to guide the search, or evaluating the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attribute or attribute values into different levels of abstraction.

4) Data mining engine: This is essential to the data mining system and ideally consists of a set of modules for tasks such as characterization, association, classification, cluster analysis, and evolution and deviation analysis.

5) Pattern evaluation module: This component typically employs interest measures and interacts with the data mining modules so as to focus the search towards interesting patterns.

6) Graphical user interface: This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, performing exploratory data mining based on the intermediate data mining results.

2.2 Process of Data Mining

The process of knowledge discovery has following steps:

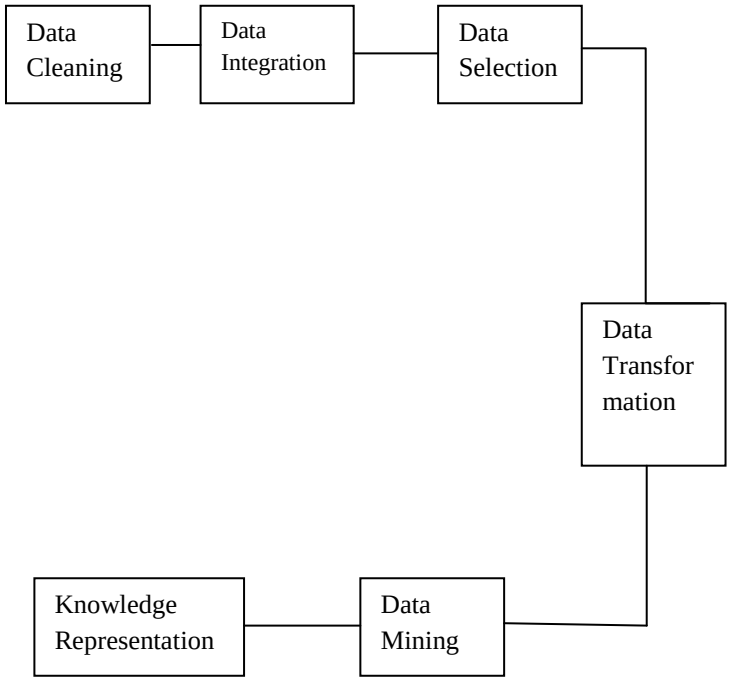

Figure 2: Process of data mining system

1) Data Cleaning: It can also be termed as data cleansing. It is a phase wherein noise data and irrelevant data are removed from the collection [2].

2) Data Integration: In this step, multiple data sources that are heterogeneous can be combined in a common source.

3) Data Selection: In this phase the data relevant to the analysis is decided on and retrieved from the data collection.

4) Data Transformation: It can also be known as data consolidation and it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.

5) Data Mining and Pattern Evaluation: Here, skilful techniques are applied to extract patterns potentially useful. Pattern Evaluation: In this phase strictly interesting patterns representing knowledge are identified with respect to the given measures.

6) Knowledge Representation: This is the final phase in which the discovered knowledge is visually represented to the user. This is an essential step which uses visualization techniques to help users understand and interpret the data mining results.

3. METHOD: CLASSIFICATION

Classification is a data mining technique that assigns items in order to target classes. To accurately predict the target class for each case of data is main goal of classification. For example, a classification model could be used to identify loan applications as low, medium or high credit risk. The task of classification begins with the data set for which the classes are known. Classifications are discrete and do not imply order.

The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values of the target. For example, high credit rating or low credit rating. Multi class have more than two target values. For example, high credit rating, low credit rating and medium credit rating.

In training process, classification algorithm finds the relation between the value of predictor and target. Different classification algorithm uses different techniques for finding the relationships. Classification model are tested by comparing the predicted values in test data. The data for classification is usually divided into two data sets: one for building the model and other for testing the model. Classification has many applications in customer segmentation, business modelling, marketing, credit analysis, biomedical and drug response modelling. We have many classification techniques like naïve Bayesian classification, Rule based classification, Classification by backpropogation, and Support vector machine, classification by decision tree induction, K-nearest neighbors, Rough set approach, and Fuzzy set approach. We will discuss Classification by decision tree induction and K- nearest neighbors using sequential approach as well as parallel approach.

4. PARALLEL APPROCHES FOR DATA MINING

Many scientific and computer related and large problems can be bettlerly solved using parallel programming. The memory limits faced by serial classifiers and need of classifying larger data set in shorter time, make classification algorithm to solve the task using parallel approach. The parallel formulation, must address issue of efficiency and scalability both in terms of memory requirements and parallel run time. Data mining can be executed in a highly parallel environment over multiple processors [6].

Modern programming languages are also structured so as to efficiently utilize novel architectures. There exist dedicated parallel programming paradigms for parallelizing the algorithms over multiprocessor and networked systems. OPENMP and MPI are used to achieve shared and distributed memory parallelization.

CUDA is a programming language that is designed for parallel programming on NVIDIA GPU [7]. In CUDA, thread access different memories of GPU. CUDA offers a data parallel programming model.

General purpose programming can also be done on the GPU where multi cores can be used for highly parallel processing. Many data mining algorithms have been specifically designed in CUDA and they show drastic improvement in performance. Parallel programming is incomplete without discussing the most recent approach called MAP Reduce. It can process large sized data in highly parallel manner [8]. Map Reduce was introduced by Google in 2004. Map Reduce has become the most popular framework for mining-large scale datasets in parallel as well as distributed environment. Different computing environments' require different programming paradigms depending upon the problem type. As data mining techniques are data and compute intensive both, it can be exploited better by using any one or combination of parallel programming approaches given in the table [9].

MPI	**OPENMP**	**CUDA**	**Map Reduce**
Framework for distributed memory parallelism.	Framework for threaded parallelism.	A parallel programming model for multiprocessing environment for GPUs.	Multi-threaded frameworks. Threads assigned for Map or reduce task.
Each has own private memory.	Shared memory.	Both shared and private memory.	Map task run on slave nodes and Reduce task work on slave nodes.
Multiple task run concurrently	Multiple threads run concurrently	Multiple light weight threads run concurrently on light weighted GPU	Library expresses two functions: Map And Reduce
Message based : send and receive	pragma omp directives	Kernel functions runs on GPUs	Based on key-value pair
On distributed network	On multicore processor	Specially designed for GPUs.	On multicore CPU, GPU, Grids and on Clouds.

Flexible and expressive.	Easier to program and debug than MPI.	Based on C- language.	Used when data size is too large.
Each processor has its own local variable.	Directives can be added.	A kernel functions has its own local variables.	Map task is highly scalable.

5. CLASSIFICATION BY DECISION TREE INDUCTION

5.1 Synchronous Tree Construction Approach

Classification by decision tree induction learning method is commonly used in data mining. To create a model that predicts the value of target variable based on some input values is main goal of this method. Each interior node Corresponds to one input value, the figure shows that concept. Each leaf node represents the value of the target variable. The input variable's value is represented by the path from root to leaf node.

Decision tree algorithm is the function of recursion. First, an attribute is selected as a root node. In order to create most efficient tree, the root node must be split the data properly. Each split attempt to pair down the actual data until they all have the same classification. The best split is the one that provides the most information gain. The basic algorithm for the decision tree is the greedy algorithm [5]. Greedy algorithm constructs the decision tree in a top-down recursive, divide and conquer manner.

The algorithm for decision tree induction for implementing it is summarized as follows.

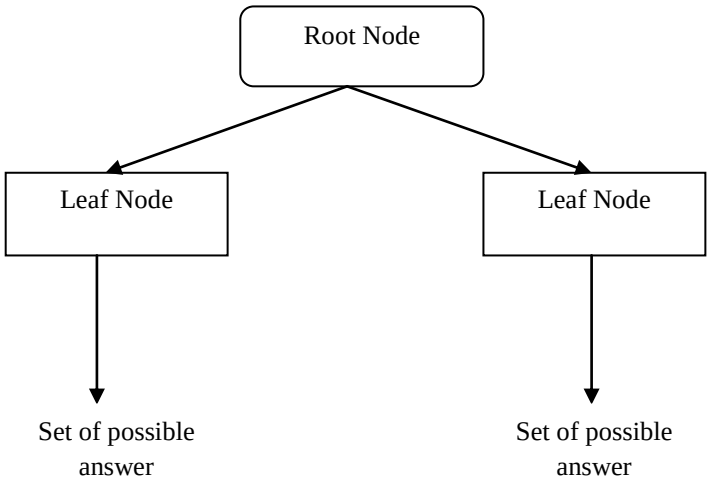

Figure 3: Classification by decision tree approach

Step 1: The algorithm operates over set of training instances, C.

Step 2: If all the instances in C are in class P, create a node P and then stop, or else select a feature or attribute F and you can create a decision node.

Step 3: Partition the training instances in C into subsets according to the values of V.

Step 4: Apply the algorithm recursively to each of subsets C.

This algorithm usually follows a greedy strategy that grows the decision tree by making series of locally optimum decisions about which attribute to use for partitioning the data.

5.2 Partitioned Tree Construction Approach

In this approach, different processors work on different parts of the classification tree. Consider the case where a group of processors Pn, correspond to expand the node n. the algorithm has following to steps:

Step 1: Processors in Pn cooperate to expand the node as explained previously.

Step 2: Once the node n is expanded into successor nodes, n1, n2,……, nk, then the processor group Pn, is partitioned, and the successor nodes are assigned to processors as follows.

1) Partition the successor nodes into Pn groups such that the total number of training cases corresponding to each node group is roughly equal. Assign each processor to one node group.

2) Shuffle the training data such that each processor has data items that belong to the nodes it is responsible for.

3) Now the expansion of the sub trees rooted at a node group proceeds completely independently at each processor as in the serial algorithm.

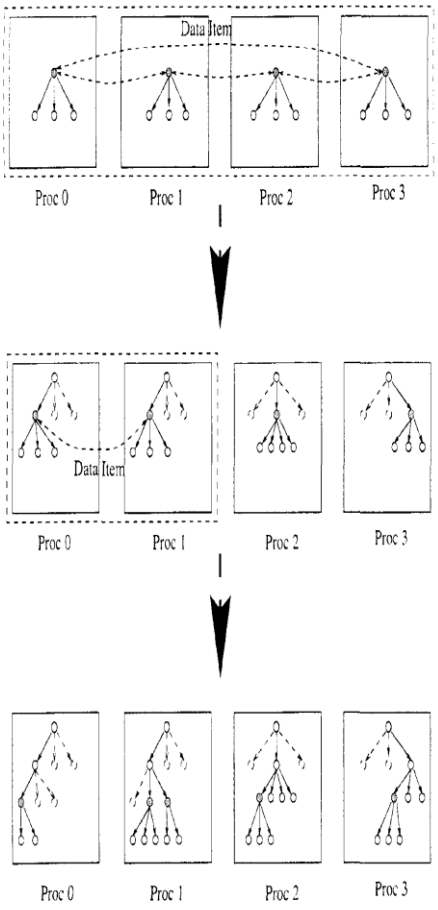

Figure 4: Parallel Classification by Decision Tree Algorithm [9]

Figure shows an example. First, all four processor expand the root node as they do in synchronous tree approach. In middle figure, the set of four processors are partitioned into three parts. The leftmost child is assigned to processor 0 and 1, while the other nodes are assigned to processor 2 and 3. Now the set of processors proceed independently to expand these nodes which are already assigned. Processor 2 and 3 proceed to expand their part of the tree using the serial algorithm. The group containing processor 0 and 1 splits the left most child into three nodes. These three new nodes are partitioned into two parts as shown in last figure; the left most node is assigned to processor 0, while other two are assigned to processor 1. From here, processors 0 and 1 also independently work on their respective sub trees.

6. CUKNN: A CUDA-BASED PARALLEL IMPLEMENTATION OF KNN

6.1 K-Nearest Neighbor

KNN algorithm is a method for classifying an object based on the closest reference object. KNN is a lazy leaner. Here a query object is classified by a majority vote of its k nearest neighbors in the reference objects. Given an unknown object p, a KNN classifier searches the reference dataset for the k objects that are closest to p, and then, p is assigned to the majority class among the k-nearest neighbors. "Closeness" is usually defined as a distance metric, such as Euclidian distance, Cosine distance etc. The most time consuming part of KNN is distance calculation component and sorting component. Therefore, the work focuses on accelerating these two components:

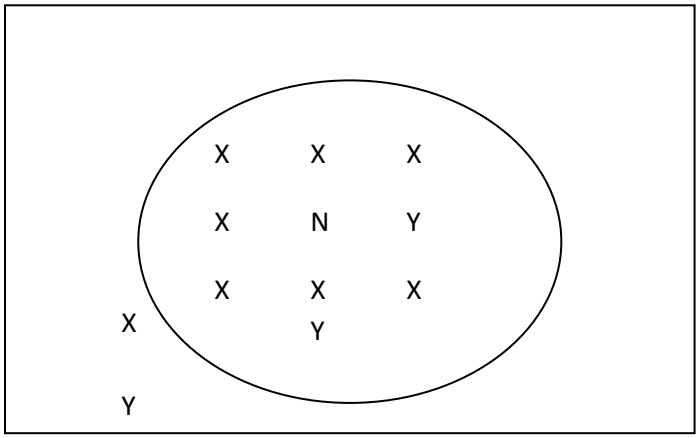

Figure 5: k-nearest neighbor. n is a new case. It would be assigned to the class x because the seven x's within the ellipse outnumber the two y's

1) Distance Calculation Kernel

The computation of the distance can be fully parallelized since the distances between pair of tuples are independent. This property makes KNN suitable for a GPU parallel implementation. After transferring the data from CPU to GPU, each thread performs the distance calculation between the query object and a reference object. Threads in a common block share the reference objects with others. Since the number of objects is large, a large number of threads and blocks are launched in this kernel. In distance calculation kernel, both reference data and query data are loaded from global memory into shared memory. Each Stream Processor in a block fetches data from shared memory [4].

2) Sorting Kernel

After calculating distance between query object, p, and reference object, sorting is performed to find KNN of p. The distances calculated by threads in common block are stored in shared memory. Threads ti takes care of one distance di. By comparing distance calculated by other threads, ti obtains the rank of di. All the threads in common block generate such task simultaneously, called local-k nearest neighbours of p. We use only one thread t to find the global-k nearest neighbors across all the blocks from the local-k neighbors on each block. We launch m blocks and each block stores k shortest distance in ascending order. In first iteration, thread t selects the global shortest one from the m local-k neighbors. Multiple times this step is repeated until k global nearest neighbors of p are selected. This is shown in the figure. Once the global-k nearest neighbors is obtained, it is easy to figure out the class label of p.

6.2 Multiple Query Objects

The number of query objects to be classified is usually more than one in real application. The performance of performing classification for multiple query objects in a parallel manner is definitely superior to the sequential manner. Method is as follows:

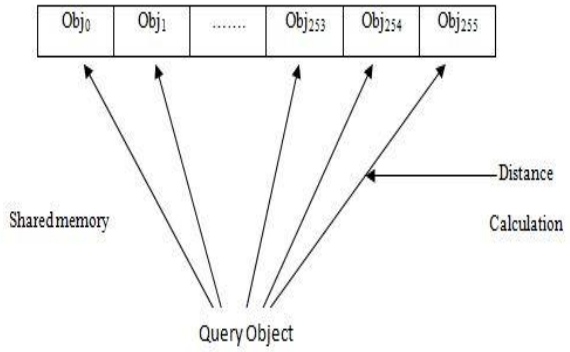

Figure 6: Parallel Approach for KNN

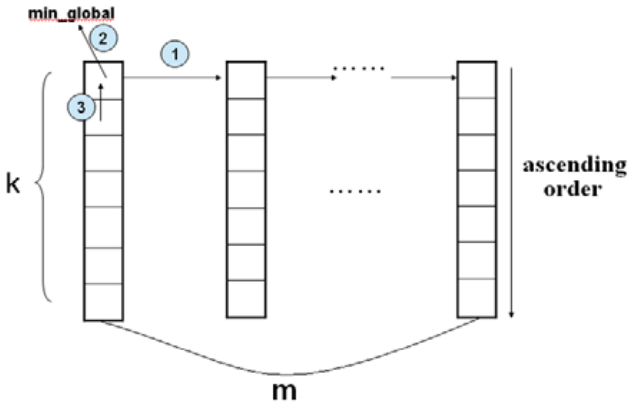

Figure 7: Global- KNN from m local –KNN [9]

Distance calculation kernel. In addition to a portion of reference objects, all the query objects are loaded into the shared memory of every block in the GPU kernels from the CPU. Then, the kernel is launched where each thread performs the distance calculation between the query objects and a reference object. Thus, at the end of this kernel, the distances between each query object and the reference object is calculated, respectively.

Sorting kernel. For a given query object, the sorting process is applied. Assume q query objects to be classified, q threads are used to generate the global-k nearest neighbors with one query per thread. In this way, the sorting process of the q query objects is performed in parallel. Notice that all the operations are performed in shared memory, which is highly efficient.

7. CONCLUSION AND FUTURE WORK

Data mining has become more relevant today with the increase in the amount of data generated every minute. Business repeated data, medical data, data on banking sectors and data on social media are really increasing now days. With issues like increase in size, data distribution, unstructured data, cleaning and pre-processing and is an open challenge. To generate more certain, precise and accurate system results is the main goal of the classification. Many methods have already been suggested for the creation of ensemble of classifiers. Several of the classification methods produce a set of interacting logic that best predict the phenotype. Using parallel approach we can reduce the amount of time taken by an algorithm. In this paper we have surveyed the classification techniques like classification by decision tree algorithm and KNN using both sequential and parallel approaches. A parallel technique takes less timing for the large size of the data. In future we can develop a parallel programming model that particularly predict the class using CUDA or any other parallel programming approach that takes short time for the large size of the data, even with the images.

8. ACKNOWLEDGEMENT

This paper is carried out with the full support from Bhaskaracharya Institute for Space Applications and Geo-informatics and the director of institute Mr T. P. Singh. I am also thankful to all the members of the institute for supplying the precious data and resources.

9. REFERENCES

[1] Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, 2nd edition.

[2] Kesavaraj, G.; Sukumaran, S., "A study on classification techniques in data mining," Computing, Communications and Networking Technologies (ICC-CNT),2013 Fourth International Conference on , vol., no., pp.1,7, 4-6 July 2013

[3] Kotecha, R.; Ukani, V.; Garg, S., "An empirical analysis of multiclass classification techniques in data mining," Engineering (NUiCONE), 2011 Nirma University International Conference on , vol., no., pp.1,5, 8-10 Dec. 2011 doi: 10.1109/NUiConE.2011.6153244

[4] Shenshen Liang; Ying Liu; ChengWang; Liheng Jian, "Design and evaluation of a parallel k-nearest neighbor algorithm on CUDA-enabled GPU," Web Society (SWS), 2010 IEEE 2nd Symposium on , vol., no., pp.53,60, 16-17 Aug. 2010 doi: 10.1109/SWS.2010.5607480

[5] Shraddha Masih and Sanjay Tanwani, “Data Mining Techniques in Parallel and Distributed Environment- A Comprehensive Survey”, IJETAE, Volume 4, Issue 3, March 2014.

[6] Wang, Lizhe, et al. "G-Hadoop: MapReduce across distributed data centers for data-intensive computing." Future Generation Computer Systems 29.3 (2013): 739-750

[7] Nickolls, John, et al. "Scalable parallel programming with CUDA." Queue 6.2 (2008): 40-53.

[8] K. Bhaduri, R. Wolf, C. Giannella, and H. Kargupta. “Distributed decision-tree induction in peer-to-peer systems.” Stat. Anal. Data Min., 1(2):85–103, 2008.

[9] Yike Guo and R. Grossman, “HIGH PERFORMANCE DATA MINING Scaling Algorithms, Applications and Systems”, A Special Issue of DATA MINING AND KNOWLEDGE DISCOVERY, Volume 3, No. 03(1999).

[10] Jhummerwala Abdul, M. B. Potdar, Prashant Chauhan, “Parallel and Distributed GIS for processing Geo-data: An Overview”, International Journal of Computer Applications Volume 106, issue 16