CLASSIFICATION OF BIG DATA USING SUPPORT VECTOR MACHINE

[1]Ms.Poonam Bonde, [2] Mr. Sachin Barahate
[1]P.G Student, [2]Assistent Professor in I.T. Department
[1]Student of YTGOIFOE, Mumbai, India
[2]Padmabhushan Vasantdada Patil Pratishthan's College Of Engineering, Sion, Mumbai, India

Abstract—In this project, Support Vector Machine is used for classifying big data. Support Vector Machine is supervised learning algorithm currently used in machine learning for classifying large data sets. Here, Stock market data is used for classification. Percentage gain & date are considered as feature vectors & data is classified based on these feature vectors in two dimensional space. As SVM classifies data into two halves, the high gain data points for that date will go on positive side of hyper plane & others will go on negative side of hyperplane. Some points may lie on margin. This classification is useful for predicting beneficial investments. All positive side data can be considered as low risk investment side or beneficial investment which can be preferred. As, stock market data file may consist of millions of records it constitutes a big data which we will try to handle by using Hadoop.
We will modify the linear SVM classifier to incremental, proximal approach which will be capable of retrieving old data and adding new data. By combining incremental SVM with hadoop Map Reduce, big data can be classified.

Index Terms—Support Vector Machine, Supervised Learning, Big Data, Hadoop, MapReduce, Classification.

I. INTRODUCTION

What is Big Data:

As its name suggests big data means huge amount of data which maybe in structured or in unstructured format. Size of Big data maybe device specific. Eg. for gmail application file of size exceeding 25 mb for transfer may be big data as it require google drive for transferring. For some applications handling gigabytes or terabytes of data is not an issue. The size of data at which processing for an application becomes difficult is considered as big data for that application as data processing capacity is different for every device.

General Characteristics of big data are:

1. Volume
2. Velocity
3. Variability
4. Variety
5. complexity

As We know, sources of big data generation are increasing day by day. So the problem of processing and classification of big data is becoming a major issue. Because of its complex, heterogeneous and distributed nature, knowledge discovery from big data has became challenging. Classification may help to find relevant information from huge data. Hadoop is well known for handling big data efficiently. Support Vector Machine is a supervised learning algorithm currently used in machine learning for classifying large data sets. By combining these two techniques, complex big data can be easily classified which can be used for analysis and research.

Sources of Big Data:

Big data comes from various sources. It comes from:

1. **Social network profiles**—Tapping user profiles from Facebook, LinkedIn, Yahoo, Google, and specific-interest social or travel sites, to cull individuals' profiles and demographic information, and extend that to capture their hopefully-like-minded networks. (This requires a fairly straightforward API integration for importing pre-defined fields and values – for example, a social network API integration that gathers every B2B marketer on Twitter.)
2. **Social influencers**—Editor, analyst and subject-matter expert blog comments, user forums, Twitter & Facebook "likes," Yelp-style catalog and review sites, and other review-centric sites like Apple's App Store, Amazon, ZDNet, etc. (Accessing this data requires Natural Language Processing and/or text-based search capability to evaluate the positive/negative nature of words and phrases, derive meaning, index, and write the results).
3. **Activity-generated data**—Computer and mobile device log files, aka "The Internet of Things." This category includes web site tracking information, application logs, and sensor data – such as check-ins and other location tracking – among other machine-generated content. But consider also the data generated by the processors found within vehicles, video games, cable boxes or, soon, household appliances. (Parsing technologies such as those from Splunk or Xenos help make sense of these types of semi-structured text files and documents.)
4. **Software as a Service (SaaS) and cloud applications**—Systems like Salesforce.com, Netsuite, SuccessFactors, etc. all represent data that's already in the Cloud but is difficult to move and merge with internal data. (Distributed data integration technology, in-memory caching technology and API integration work may be appropriate here.)
5. **Public**—Microsoft Azure Marketplace/DataMarket, The World Bank, SEC/Edgar, Wikipedia, IMDb, etc. – data that is publicly available on the Web which may enhance the types of analysis able to be performed. (Use the same types of parsing, usage, search and categorization techniques as for the three previously mentioned sources.)
6. **Hadoop MapReduce application results**—The next generation technology architectures for handling and parallel parsing of data from logs, Web posts, etc., promise to create a new generations of pre- and post-processed data. We foresee a ton of new products that will address application use cases for any kinds of Big Data – just look at the partner lists of Cloudera and Hortonworks. In fact, we won't be surprised if layers of MapReduce applications blending everything mentioned above

(consolidating, "reducing" and aggregating Big Data in a layered or hierarchical approach) are very likely to become their own "Big Data".

7. **Data warehouse appliances**—Teradata, IBM Netezza, EMC Greenplum, etc. are collecting from operational systems the internal, transactional data that is already prepared for analysis. These will likely become an integration target that will assist in enhancing the parsed and reduced results from your Big Data installation.
8. **Columnar/NoSQL data sources**—MongoDB, Cassandra, InfoBright, etc. – examples of a new type of map reduce repository and data aggregator. These are specialty applications that fill gaps in Hadoop-based environments, for example Cassandra's use in collecting large volumes of real-time, distributed data.
9. **Network and in-stream monitoring technologies**—Packet evaluation and distributed query processing-like applications as well as email parsers are also likely areas that will explode with new startup technologies.
10. **Legacy documents**—Archives of statements, insurance forms, medical record and customer correspondence are still an untapped resource. (Many archives are full of old PDF documents and print streams files that contain original and only systems of record between organizations and their customers. Parsing this semi-structured legacy content can be challenging without specialty tools like Xenos.)[1]

Fig 1. Sources of big data

Major Challenges

Challenges with respect to tiers [2]

Tier I: Big data mining platform:

A computing platform is needed to have efficient access to, at least, two types of resources: data and computing processors. For small scale data mining tasks, a single desktop computer, which contains hard disk and CPU processors, is sufficient to fulfill the data mining goals. For medium scale data mining tasks, data are typically large and cannot be fit into the main memory. Common solutions are to rely on parallel computing or collective mining to sample and aggregate data from different sources and then use parallel computing programming to carry out the mining process. For Big Data mining, because data scale is far beyond the capacity that a single personal computer (PC) can handle, a typical Big Data processing framework will rely on cluster computers with a high-performance computing platform, with a data mining task being deployed by running some parallel programming tools, such as MapReduce or Enterprise Control Language (ECL), on a large number of computing nodes (i.e., clusters).

Tier II: Big Data Semantics and Application Knowledge:

Semantics and application knowledge in Big Data refer to numerous aspects related to the regulations, policies, user knowledge, and domain information. The two most important issues at this tier include

1) data sharing andprivacy; and

2) domain and application knowledge.

Tier III: Big Data Mining Algorithms:

- Local Learning and Model Fusion for Multiple Information Sources
- Mining from Sparse, Uncertain, and Incomplete Data

Mining Complex and Dynamic Data

In this project, we used stock market data as a source of big data. It is classified into two dimensional space. It can be furher classified more effectively into n dimensional space.

II. LITERATURE REVIEW

Classification is a learning function that maps a given data item into one of several predefined classes:

Support Vector Machines (SVM) are classification and regression methods which have been derived from statistical learning theory (Vapnik, 1995). The concept is based on optimal linear separating hyperplane that is fitted to the training patterns of two classes within a multi-dimensional feature space. The optimization problem that has to be solved relies on structural risk minimization and is aiming at a maximization of the margins between the hyperplane and closest training samples. If the two classes are non-separable, SVMs employ the kernel trick where a positive definite kernel function is used to map the input data into a high dimensional transformed feature space. [3]

In 1936, R. A. Fisher suggested the first algorithm for pattern recognition (Fisher 1936).

Aronszajn (1950) introduced the 'Theory of Reproducing Kernels'.

In 1957 Frank Rosenblatt invented a linear classifier called the perceptron (the simplest kind of feedforward neural network), see Rosenblatt (1962).

Vapnik and Lerner (1963) introduce the Generalized Portrait algorithm (the algorithm implemented by support vector machines is a nonlinear generalization of the Generalized Portrait algorithm).
Aizerman, Braverman and Rozonoer (1964) introduced the geometrical interpretation of the kernels as inner products in a feature space.
Vapnik and Chervonenkis (1964) further develop the Generalized Portrait algorithm.
Cover (1965) discussed large margin hyperplanes in the input space and also sparseness.
Similar optimization techniques were used in pattern recognition by Mangasarian (1965).
The use of slack variables to overcome the problem of noise and nonseparability was introduced by Smith (1968).
Duda and Hart (1973) discuss large margin hyperplanes in the input space.
The field of 'statistical learning theory' began with Vapnik and Chervonenkis (1974) (in Russian).
SVMs can be said to have started when statistical learning theory was developed further with Vapnik (1979) (in Russian).
Wapnik and Tscherwonenkis (1979) wrote a German translation of Vapnik and Chervonenkiss 1974 book.
Vapnik (1982) wrote an English translation of his 1979 book.
See also the PhD thesis by Hassoun (1986) for related early work.
Several statistical mechanics papers (for example Anlauf and Biehl (1989)) suggested using large margin hyperplanes in the input space.
Poggio and Girosi (1990) and Wahba (1990) discuss the use of kernels.
Bennett and Mangasarian (1992) improved upon Smith's 1968 work on slack variables.
SVMs close to their current form were first introduced with a paper at the COLT 1992 conference (Boser, Guyon and Vapnik 1992).
In 1995 the soft margin classifier was introduced by Cortes and Vapnik (1995); in the same year the algorithm was extended to the case of regression by Vapnik (1995) in The Nature of Statistical Learning Theory.
The papers by Bartlett (1998) and Shawe-Taylor, et al. (1998) gave the first rigorous statistical bound on the generalisation of hard margin SVMs.
Shawe-Taylor and Cristianini (2000) gave statistical bounds on the generalisation of soft margin algorithms and for the regression case.[4]

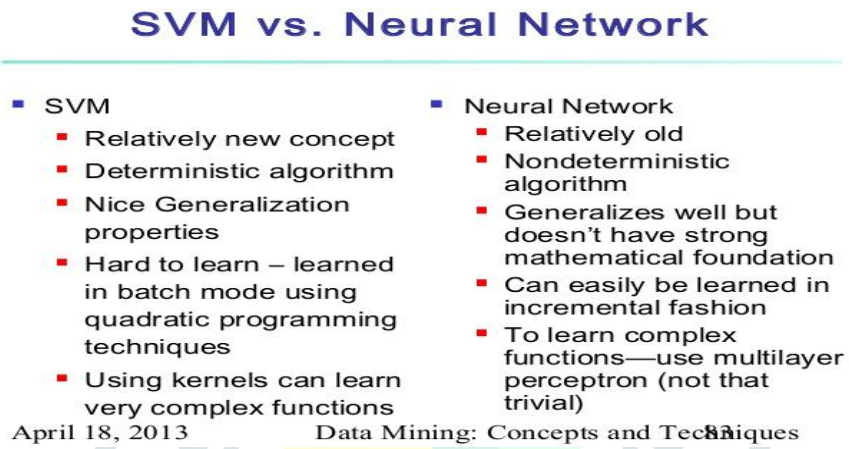

Fig 2. SVM vs Neural Network[5]

III. FRAMEWORK OF PROPOSED APPROACH

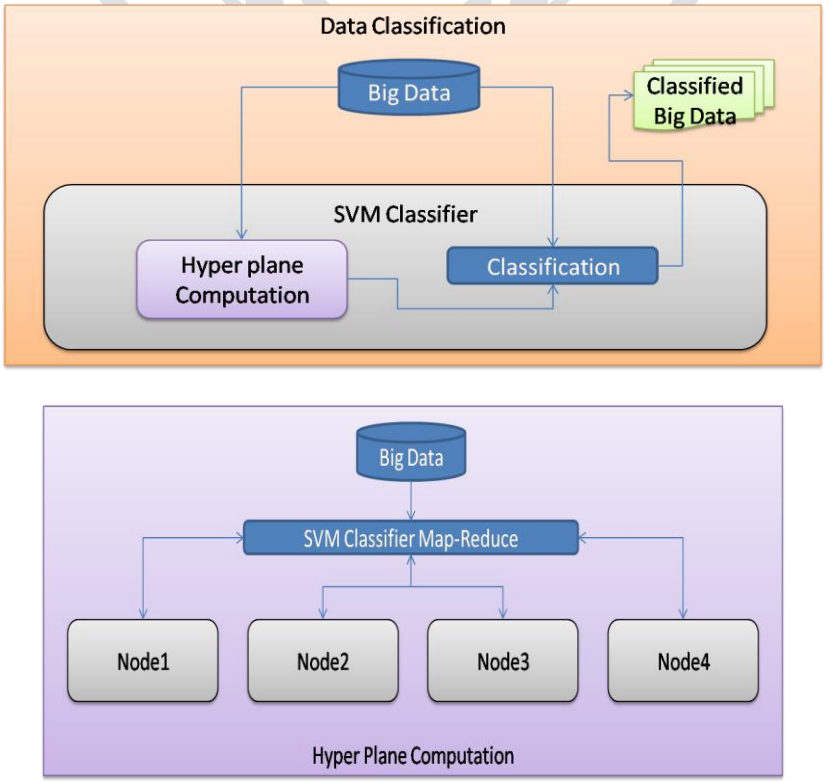

Fig 3. Proposed system

Steps:

1. Installation of Hadoop
2. Start Hadoop by using start-all.sh
3. Prepare input file of big data, here we are using stock market data for classification.
4. Decide feature vectors and give this file as input to the Hadoop map reduce and incremental SVM.
5. To compute the values of $\begin{bmatrix}\omega\\ \gamma\end{bmatrix}$ and in turn classify data in a file, we are creating a script that will solve the equation by using incremental method.
6. The incremental method will use the map-reduce functionality and will create m number of maps for the m x n matrix and these maps will be passed to the nodes by Hadoop system thus processing them parallel as well as incremental hence improving the turnaround time for SVM to classify a given set of Big data.

Basic idea of SVM:

Classification using support vector machine involves construction of an optimal hyper plane which separates a plane into two halve spaces. The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying data that maximizes the margin of the hyper plane i.e. the distance from the hyper plane to the nearest point of each patterns.

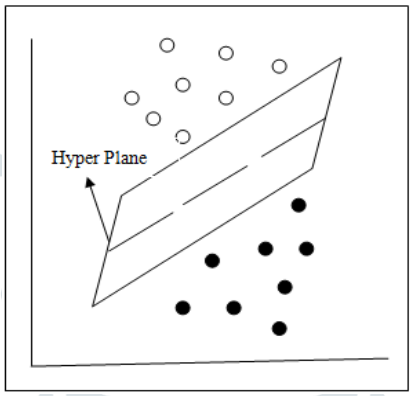

Fig 4: A Hyper Plane

Applications of SVM:

1. Text and hypertext classification
2. Medical Science i.e classifying cancer data, Severity of disease categorization
3. Bioinformatics i.e, Protein classification
4. Image Classification
5. Handwritten character recognition

Hadoop Map- Reduce:

MapReduce is a framework with programs can be written to process huge volumes of data, in parallel, on large clusters of commodity hardware in a reliable manner. It involves two important tasks i.e Map & Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into key/value pairs i.e tuples. reducer task, takes the output from a map as an input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map job.

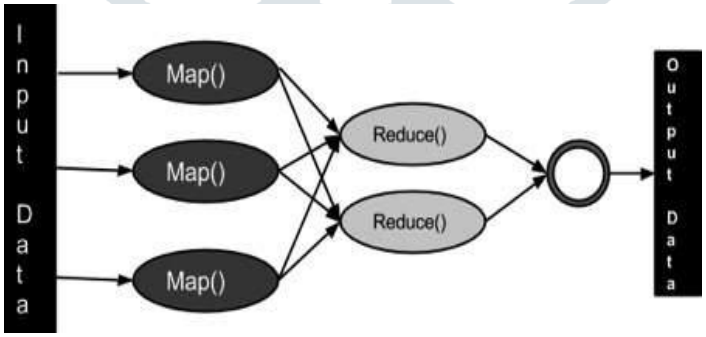

Fig 5. Map-Reduce Algorithm

Algorithm (Linear Proximal SVM):

Linear Proximal SVM Given m data points in R^n represented by the m × n matrix A and a diagonal matrix D of ±1 labels denoting the class of each row of A, we generate the linear classifier

$$\text{sign}(x'w - \gamma) = 1, \text{ then } x \in A+ \\ = -1, \text{ then } x \in A- \Big\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

as follows:

(i) Define E by E = [A − e] where e is an m × 1 vector of ones. Compute $\begin{bmatrix}\omega\\ \gamma\end{bmatrix} = (I/\nu + E'E)^{-1}E'De$ for some positive ν. Typically ν is chosen by means of a tuning (validating) set.

(ii) Classify a new x by using (1) and the solution $\begin{bmatrix}\omega\\ \gamma\end{bmatrix}$ from Step (i) above.

Algorithm (Linear Incremental SVM):

Given m records & n features in R^n represented by m x n matrix A and Diagonal matrix D of + or – 1 labels denoting the class of each row of A, we generate an incremental linear classifier by retiring old data represented by the sub matrix $E^1 \in R^{m^1 \times (n+1)}$ of E = [A -e] and a corresponding diagonal sub matrix $D^1 \in R^{m^1 \times m^1}$ of D of ± 1 and adding new data represented by a new matrix $E^2 \in R^{m^2 \times (n+1)}$ and corresponding diagonal matrix $D^2 \in R^{m^2 \times m^2}$ of D of ± 1 as follows:

$$sign(z'(\frac{I}{\nu} + E'E - E^{1'}E^1 + E^{2'}E^2)^{-1}(E'De - E^{1'}D^1e + E^{2'}D^2e)) \begin{cases} = 1, & then\ x \in A+, \\ = -1, & then\ x \in A-. \end{cases}$$

Hence, we will use this property of linear incremental SVM to prepare map-reduce and solve it by using Hadoop's Map Reduce[6].

IV. CONCLUSION

Incremental SVM can be combined with hadoop map reduce to simultaneously classifying data and handling big data effectively. Big data files can be sent to multiple nodes having hadoop and SVM for reducing processing time of classification. Here, we have implemented this on single node cluster and in two dimensional space. But it can be implemented in N dimensional space.

It has returned following support vectors which can be used to classify the data by using sign(x′w − γ) for each new record. By leveraging hadoop's inherent capability of managing big data files on multiple nodes, the processing time can be reduced further.

REFERENCES

[1] http://www.zdnet.com/article/top-10-categories-for-big-data-sources-and-mining-technologies/

[2] X. Wu, X. Zhu, G. Q. Wu and W. Ding, "Data mining with big data," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97-107, Jan. 2014. doi: 10.1109/TKDE.2013.109

[3] http://shodhganga.inflibnet.ac.in/bitstream/10603/28805/8/08_chapter%202.pdf

[4] http://www.svms.org/history.html

[5] https://www.slideshare.net/error007/06-19079438

[6] Incremental Support Vector Machine Classification Glenn Fung∗ and Olvi L. Mangasarian