Analyzing of Social Network Data using Big Data and Tools

Nilamadhab Mishra
Department of CSE
Templecity Institute of Technology & Engineering
Bhubaneswar, India

Sarojananda Mishra
Department of CSE
IndiraGandhi Institute of Technology
Dhenkanal, India

***Abstract*— Streaming and extracting useful information and knowledge through data analysis in real time is becoming complex, fastest and more efficient techniques are require. This paper discusses on the current and future trends of mining evolving data streams using big data concepts. The work also include briefing on structured and unstructured data and the challenges that the field will have to overcome during the coming.**

Keywords*— *Data Analysis, structured data, unstructured data, Big Data.

I. INTRODUCTION

At the present time, the quantity of data that is either created or acquired in every two days is estimated to be more than 5 exabytes. This amount of data is similar to the amount of data created from the dawn of time up until 2003. Moreover, it was estimated in 2007 that it was not possible to store all the data that are being produced. Thus storing and compiling the massive amount of data opens new challenging discovery tasks.

Data stream real time analytics are needed to manage the data currently generated, at an ever increasing rate, from such applications as: sensor networks, measurements in network monitoring and traffic management, log records or click- streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts and others [5]. In fact, all data generated can be considered as streaming data or as a snapshot of streaming data, since it is obtained from an interval of time.

In the data stream model, data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time. Consequently, data streams pose several challenges for data mining algorithm design. First, algorithms must make use of limited resources (time and memory). Second, they must deal with data whose nature or distribution changes over time.

One needs to deal with resources in an efficient and low-cost way. Green computing is the study and practice of using computing resources efficiently. The main approach to green computing is based on algorithmic and hardware efficiency. In data stream mining, earlier studies focused in three main dimensions:

- Accuracy
- Amount of space and
- Time required to learn from training examples and to predict

These dimensions are typically interdependent: adjusting the time and space used by an algorithm can influence accuracy. By storing more pre-computed information, such as look up tables, an algorithm can run faster at the expense of space. An algorithm can also run faster by processing less information, either by stopping early or storing less, thus having less data to process. The more time an algorithm has, the more likely it is that accuracy can be increased.

The issue of the measurement of three evaluation dimensions simultaneously has led to another important issue in data stream mining, namely estimating the combined cost of performing the learning and prediction processes in terms of time and memory. As an example, several rental cost options exist:

A. Big Data: The need of its Emergence

There is a need for radically new approaches to research data modeling. Current data models (relational model) and management systems (relational database management systems) were developed by the database research community for business/commercial data applications. Research data has completely different characteristics and thus the current database technology is unable to handle it effectively.

There is a need for data models and query languages that:

1) More closely match the data representation needs of the several scientific disciplines;
2) describe discipline-specific aspects (metadata models);
3) represent and query data provenance information;
4) represent and query data contextual information;
5) represent and manage data uncertainty;
6) represent and query data quality information.

B. Data management challenges

There is a clear need for extremely large data processing. This is especially true in the area of scientific data management where, in the near future, we expect data inputs in the order of multiple Petabytes. However, current data management technology is not suitable for such data sizes. In the light of such new database applications, we need to rethink some of the strict requirements adopted by database systems in the past. For instance, database management systems (DBMS) see database queries as contracts carved in stone that require the DBMS to produce a complete and correct answer, regardless of the time and resources required. While this behavior is crucial in business data management, it is

counterproductive in scientific data management. With the explorative nature of scientific discovery, scientists cannot be expected to instantly phrase a crisp query that yields the desired (but a priori unknown) result, or to wait days to get a multi-megabyte answer that does not reveal what they were looking for. Instead, the DBMS could provide a fast and cheap approximation that is neither complete nor correct, but indicative of the complete answer. In this way, the user gets a 'feel' for the data that helps him/her to advance his/her exploration using a refined query.

The challenges faced include catching the user's intention and providing the users with suggestions and guidelines to refine their queries in order to quickly converge to the desired results, but also call for novel database architectures and algorithms that are designed with the intent to produce fast and cheap indicative answers rather than complete and correct answers.

C. Data Tools challenges

Currently, the available data tools for most scientific disciplines are inadequate. It is essential to build better tools in order to improve the productivity of scientists. There is a need for better computational tools to visualize, analyze, and catalog the available enormous research datasets in order to enable data-driven research.

Scientists need advanced tools that enable them to follow new paths, try new techniques, build new models and test them in new ways that facilitate innovative multidisciplinary / interdisciplinary activities and support the whole research cycle.

D. Features of Big Data

To clarify matters, the three Vs of *volume, velocity and variety* are commonly used to characterize different aspects of big data. They're a helpful lens through which to view and understand the nature of the data and the software platforms available to exploit them. Most probably you will contend with each of the Vs to one degree or another.

1) Volume: The benefit gained from the ability to process large amounts of information is the main attraction of big data analytics. Having more data beats out having better models: simple bits of math can be unreasonably effective given large amounts of data. If you could run that forecast taking into account 300 factors rather than 6, could you predict demand better?

This volume presents the most immediate challenge to conventional IT structures. It calls for scalable storage, and a distributed approach to querying. Many companies already have large amounts of archived data, perhaps in the form of logs, but not the capacity to process it.

Assuming that the volumes of data are larger than those conventional relational database infrastructures can cope with, processing options break down broadly into a choice between massively parallel processing architectures — data warehouses or databases such as Green plum — and Apache Hadoop-based solutions. This choice is often informed by the degree to which the one of the other "Vs" — variety — comes into play. Typically, data warehousing approaches involve predetermined schemas, suiting a regular and slowly evolving dataset. Apache Hadoop, on the other hand, places no conditions on the structure of the data it can process.

At its core, Hadoop is a platform for distributing computing problems across a number of servers. First developed and released as open source by Yahoo, it implements the MapReduce approach pioneered by Google in compiling its search indexes. Hadoop's MapReduce involves distributing a dataset among multiple servers and operating on the data: the "map" stage. The partial results are then recombined: the "reduce" stage.

To store data, Hadoop utilizes its own distributed filesystem, HDFS, which makes data available to multiple computing nodes. A typical Hadoop usage pattern involves three stages:

- loading data into HDFS,
- MapReduce operations, and
- retrieving results from HDFS.

This process is by nature a batch operation, suited for analytical or non-interactive computing tasks. Because of this, Hadoop is not itself a database or data warehouse solution, but can act as an analytical adjunct to one.

One of the most well-known Hadoop users is Facebook, whose model follows this pattern. A MySQL database stores the core data. This is then reflected into Hadoop, where computations occur, such as creating recommendations for you based on your friends' interests. Facebook then transfers the results back into MySQL, for use in pages served to users.

2)Velocity: The importance of data's velocity — the increasing rate at which data flows into an organization — has followed a similar pattern to that of volume. Problems previously restricted to segments of industry are now Presenting themselves in a much broader setting. Specialized companies such as financial traders have long turned systems that cope with fast moving data to their advantage. Now it's our turn.

Why is that so? The Internet and mobile era means that the way we deliver and consume products and services is increasingly instrumented, generating a data flow back to the provider. Online retailers are able to compile large histories of customers' every click and interaction: not just the final sales. Those who are able to quickly utilize that information, by recommending additional purchases, for instance, gain competitive advantage. The smartphone era increases again the rate of data inflow, as consumers carry with them a streaming source of geolocated imagery and audio data.

As per the IBM researchers - It's not just the velocity of the incoming data that's the issue: it's possible to stream fast- moving data into bulk storage for later batch processing, as an example. The importance lies in the speed of the feedback loop, taking data from input through to decision. A commercial from IBM makes the point that you wouldn't cross the road if all

you had was a five-minute old snapshot of traffic location. There are times when you simply won't be able to wait for a report to run or a Hadoop job to complete.

Industry terminology for such fast-moving data tends to be either "streaming data," or "complex event processing." This latter term was more established in product categories before streaming processing data gained more widespread relevance, and seems likely to diminish in favor of streaming.

There are two main reasons to consider streaming processing. The first is when the input data are too fast to store in their entirety: in order to keep storage requirements practical some level of analysis must occur as the data streams in. At the extreme end of the scale, the Large Hadron Collider at CERN generates so much data that scientists must discard the overwhelming majority of it — hoping hard they've not thrown away anything useful. The second reason to consider streaming is where the application mandates immediate response to the data. Thanks to the rise of mobile applications and online gaming this is an increasingly common situation.

Product categories for handling streaming data divide into established proprietary products such as IBM's InfoSphere Streams, and the less-polished and still emergent open source frameworks originating in the web industry: Twitter's Storm, and Yahoo S4.

As mentioned above, it's not just about input data. The velocity of a system's outputs can matter too. The tighter the feedback loop, the greater the competitive advantage. The results might go directly into a product, such as Facebook's recommendations, or into dashboards used to drive decision- making.

It's this need for speed, particularly on the web, that has driven the development of key-value stores and columnar databases, optimized for the fast retrieval of precomputed information. These databases form part of an umbrella category known as NoSQL, used when relational models aren't the right fit.

Microsoft SQL Server is a comprehensive information platform offering enterprise-ready technologies and tools that help businesses derive maximum value from information at the lowest TCO. SQL Server 2012 launches next year, offering a cloud-ready information platform delivering mission-critical confidence, breakthrough insight, and cloud on your terms.

3) Variety: Rarely does data present itself in a form perfectly ordered and ready for processing. A common theme in big data systems is that the source data is diverse, and doesn't fall into neat relational structures. It could be text from social networks, image data, a raw feed directly from a sensor source. None of these things come ready for integration into an application.

Even on the web, where computer-to-computer communication ought to bring some guarantees, the reality of data is messy. Different browsers send different data, users withhold information, they may be using differing software versions or vendors to communicate with you. And you can bet that if part of the process involves a human, there will be error and inconsistency.

A common use of big data processing is to take unstructured data and extract ordered meaning, for consumption either by humans or as a structured input to an application. One such example is entity resolution, the process of determining exactly what a name refers to. Is this city London, England, or London, Texas? By the time your business logic gets to it, you don't want to be guessing.

The process of moving from source data to processed application data involves the loss of information. When you tidy up, you end up throwing stuff away. This underlines a principle of big data: *when you can, keep everything*. There may well be useful signals in the bits you throw away. If you lose the source data, there's no going back.

Despite the popularity and well understood nature of relational databases, it is not the case that they should always be the destination for data, even when tidied up. Certain data types suit certain classes of database better. For instance, documents encoded as XML are most versatile when stored in a dedicated XML store such as MarkLogic. Social network relations are graphs by nature, and graph databases such as Neo4J make operations on them simpler and more efficient.

Even where there's not a radical data type mismatch, a disadvantage of the relational database is the static nature of its schemas. In an agile, exploratory environment, the results of computations will evolve with the detection and extraction of more signals. Semi-structured NoSQL databases meet this need for flexibility: they provide enough structure to organize data, but do not require the exact schema of the data before storing it.

4. ADDITIONAL FEATURES OF BIG DATA

a) Variability - This is a factor which can be a problem for those who analyse the data. This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

b) Veracity - The quality of the data being captured can vary greatly. Accuracy of analysis depends on the veracity of the source data.

c) Complexity - Data management can become a very complex process, especially when large volumes of data come from multiple sources. These data need to be linked, connected and correlated in order to be able to grasp the information that is supposed to be conveyed by these data. This situation, is therefore, termed as the 'complexity' of Big Data.

E. Big Data cloud implications

Typical big data projects focus on scaling or adopting Hadoop for data processing. MapReduce has become a de facto standard for large scale data processing. Tools like Hive and Pig have emerged on top of Hadoop which make it feasible to process huge data sets easily. Hive for example transforms SQL like queries to MapReduce jobs. It unlocks data set of all sizes for

data and business analysts for reporting and greenfield analytics projects.

Data can be either transferred to or collected in a cloud data sink like Amazon's S3, e.g. to collect log files or export text formatted data. Alternatively database adapters can be utilized to access data from databases directly with Hadoop, Hive, and Pig. Qubole is a leading provider of cloud based services in this space. They provide unique database adapters that can unlock data instantly, which otherwise would be inaccessible or require significant development resource. One great example is their mongoDB adapter. It gives Hive table like access to mongoDB collections. Qubole scales Hadoop jobs to extract data as quickly as possible without overpowering the mongoDB instance.

Ideally a cloud service provider offers Hadoop clusters that scale automatically with the demand of the customer. This provides maximum performance for large jobs and optimal savings when little and no processing is going on.

I. NEW PROBLEMS: STRUCTURED AND UNSTRUCTURED CLASSIFICATION

A new important and challenging task may be the structured pattern classification problem. Patterns are elements of (possibly infinite) sets endowed with a partial order relation ≤. Examples of patterns are item sets, sequences, trees and graphs.

The structured pattern classification problem is defined as follows. A set of examples of the form (t; y) is given, where y is a discrete class label and t is a pattern.

The goal is to produce from these examples a model ^y = f(t) that will predict the classes y of future pattern examples Most standard classification methods can only deal with vector data, which is but one of many possible pattern structures. To apply them to other types of patterns, such as graphs, we can use the following approach: we convert the pattern classification problem into a vector classification learning task, transforming patterns into vectors of attributes. Each attribute denotes the presence or absence of particular sub patterns, and we create attributes for all frequent sub- patterns, or for a subset of these.

As the number of frequent sub patterns may be very large, we may perform a feature selection process, selecting a subset of these frequent sub patterns, maintaining exactly or approximately the same information.

The structured output classification problem is even more challenging and is defined as follows. A set of examples of the form (t; y) is given, where t and y are patterns.

The goal is to produce from these examples a pattern model ^y = f(t) that will predict the patterns y of future pattern examples. A way to deal with a structured output classification problem is to convert it to a multi label classification problem, where the output pattern y is converted into a set of labels representing a subset of its frequents sub patterns.

Therefore, data stream multi-label classification methods may offer a solution to the structured output classification problem.First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the file "MSW_USltr_format".

*A. **Maintaining the Integrity of the Specifications***

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

II. NEW APPLICATIONS: DYNAMIC SOCIAL NETWORK

Social network defines the set of relationships between individuals, where each individual is a social entity. The collection of ties between people and strength of those ties is defined by social network. Mathematically, the social structure made of nodes (which are generally individuals or organizations) that are tied by one or more specific types of interdependency, such as values, visions, ideas, financial exchange, friendship, sexual relationships, kinship, dislike, conflict or trade are interpreted by the term of Social Network [15].

Social networks are dynamic by nature. Ties are established, they may flourish and perhaps evolve into close relationships, and they can also dissolve quietly, or suddenly turn sour and go with a bang. These relational changes may be considered the result of the structural positions of the actors within the network considering the example, when friends of friends become friends –, characteristics of the actors ('actor covariates'), characteristics of pairs of actors ('dyadic covariates'), and residual random influences representing unexplained Influences.

A dynamic network consists of ties between actors that change over time. A foundational assumption of the models discussed in this paper is that the network ties are not brief events, but can be regarded as states with a tendency to endure over time. Many relations commonly studied in network analysis naturally satisfy this requirement of gradual change, such as friendship, trust, and cooperation. Other networks more strongly resemble 'event data', an example can be viewed as the set of all telephone calls among a group of actors at any given time point, or the set of all e-mails being sent at any given time point. While it is meaningful to interpret these networks as indicators of communication, it is not plausible to treat their ties as enduring states, although it often is possible to aggregate event intensity over a certain period and then view these aggregates as indicators of states.

Given that the network ties under study denote states, it is further assumed, as an approximation, that the changing network can be interpreted as the outcome of a Markov process [17] , i.e., that for any point in time, the current state of the network determines probabilistically its further evolution, and there are no additional effects of the earlier past. All relevant information is therefore assumed to be included in the current state. This assumption often can be made more plausible by choosing meaningful independent variables that incorporate relevant information from the past.

III. NEW TECHNIQUES: S4, HANDOOP OR STORM

A way to speed up the mining of streaming learners is to distribute the training process onto several machines. Hadoop Map Reduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes.

A Map Reduce job divides the input dataset into independent subsets that are processed by map tasks in parallel. This step of mapping is then followed by a step of reducing tasks. These reduce tasks use the output of the maps to obtain the final result of the job.

Apache S4 [7] is a platform for processing continuous data streams. S4 is designed specifically for managing data streams. S4 apps are designed combining streams and processing elements in real time. Storm from Twitter uses a similar approach.

Ensemble learning classifiers are easier to scale and parallelize than single classifier methods. They are the first, most obvious, candidate methods to implement using parallel techniques.

IV. CONCLUSION

We have discussed the challenges that in our structured and unstructured, mining evolving data streams will have to deal during the coming years. The authors have outlined the recent areas for research. These include structured classification and associated application areas as social networks. One's ability to handle many exabytes of data across many application areas in the coming days will be crucially dependent on the existence of a rich variety of datasets, techniques and software frameworks. There is no doubt that data stream mining offers many challenges and equally many opportunities as the quantity of data generated in real time is going to continue growing.

REFERENCES

[1] A. Bifet and E. Frank. Sentiment knowledge discovery in Twitter streaming data. In Proc 13th International Conference on Discovery Science, Canberra, Australia, pages 1{15. Springer,2010.

[2] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer. MOA: Massive Online Analysis http://moa.cms.waikato.ac.nz/. Journal of Machine Learning Research (JMLR), 2010.

[3] A. Bifet, G. Holmes, and B. Pfahringer. Moa-tweetreader: Real-time analysis in twitter streaming data. In Discovery Science, pages 46{60, 2011.

[4] A. Bifet, G. Holmes, B. Pfahringer, and E. Frank. Fast perceptron decision tree learning from evolving data streams. In PAKDD, 2010.

[5] J. Gama. Knowledge discovery from data streams. Chapman & Hall/CRC, 2010.

[6] B. Liu. Web data mining; Explor ing hyperlinks, contents, and usage data. Springer, 2006.

[7] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed stream computing platform. In ICDM Workshops, pages 170-177, 2010.

[8] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1-135, 2008.

[9] Mitra,A; Satapathy,S R; Paul,S, " Clustering in Social Network using Covering Based Rough Set" IEEE – Int. Conf. (IACC 2013)(available in IEEE Xplorer) Ghaziabad, India, (10.1109/IadCC.2013.6514272), Pp. 476-481, Feb 22-23, 2013.

[10] Tom A.B. Snijders, "The Statistical Evaluation of Social Network Dynamics", Sociological Methodology, January 2001.