JIMMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Title:- Stream processing

Name ……...ID no

Mohammed Usman……RU0497/08

Submitted to: Mr.Dessalgn

1. Introduction

In recent years, advances in hardware technology have facilitated the ability to collect data continuously. Simple transactions of everyday life such as using a credit card, a phone or browsing the web lead to automated data storage. Similarly, advances in information technology have lead to large flows of data across IP networks. In many cases, these large volumes of data can be mined for interesting and relevant information in a wide variety of applications.

When the volume of the underlying data is very large, it leads to a number of computational and mining challenges:

- ✓ With increasing volume of the data, it is no longer possible to process the data efficiently by using multiple passes. Rather, one can process a data item at most once. This leads to constraints on the implementation of the underlying algorithms. Therefore, stream mining algorithms typically need to be designed so that the algorithms work with one pass of the data.
- ✓ In most cases, there is an inherent temporal component to the stream mining process. This is because the data may evolve over time. This behavior of data streams is referred to as *temporal locality*. Therefore, a straightforward adaptation of one-pass mining algorithms may not be an effective solution to the task. Stream mining algorithms need to be carefully designed with a clear focus on the evolution of the underlying data.
- ✓

Another important characteristic of data streams is that they are often mined in a distributed fashion. Furthermore, the individual processors may have limited processing and memory. Examples of such cases include sensor networks, in which it may be desirable to perform in-network processing of data stream with limited processing and memory.

2. Descriptions of Basic algorithms of stream processing

2.1. LWClass Algorithm

Gaber et al have proposed Lightweight Classification techniques termed as LWClass. LWClass is based on Algorithm Output Granularity. The algorithm output granularity (AOG) introduces the first resource-aware data analysis approach that can cope with fluctuating data rates according to the available memory and the processing speed. The AOG performs the local data analysis on resource constrained devices that generate or receive streams of information.

- ❖ AOG has three stages of:-
- mining,
- adaptation and
- knowledge integration

LWClass starts with determining the number of instances that could be resident in memory according to the available space. Once a classified data record arrives, the algorithm searches for the nearest instance already stored in the main memory. This is done using a pre-specified distance threshold. This threshold represents the similarity measure acceptable by the algorithm to consider two or more data records as an entry into a matrix.

This matrix is a summarized version of the original data set. If the algorithm finds a nearest neighbor, it checks the class label. If the class label is the same, it increases the weight for this instance by one, otherwise it decrements the weight by one. If the weight is decremented down to zero, this entry will be released from the memory conserving the limited memory on streaming applications.

The algorithm output granularity is controlled by the distance threshold value and is changing over time to cope with the high speed of the incoming data elements.

The algorithm procedure could be described as follows:-

Each record in the data stream contains attribute values for $a_1, a_2, \ldots, a_n$ attributes and the class category.

1. According to the data rate and the available memory, the algorithm output granularity is applied as follows:

1.1 Measure the distance between the new record and the stored ones.

1.2 If the distance is less than a threshold, store the average of these two records and increase the weight for this average as an entry by 1. (The threshold value determines the algorithm accuracy and is chosen according to the available memory and data rate that determines the algorithm rate).

This is in case that both items have the same class category. If they have different class categories, the weight is decreased by 1 and released from memory if the weight reaches zero.

1.3 After a time threshold for the training, we come up with a matrix.

2. Using Table 3.3, the unlabeled data records could be classified as follows.According to the available time for the classification process, we choose nearest K-table entries and these entries are variable according to the time needed by the process.
3. Find the majority class category taking into account the calculated weights from the K entries. This will be the output for this classification task.

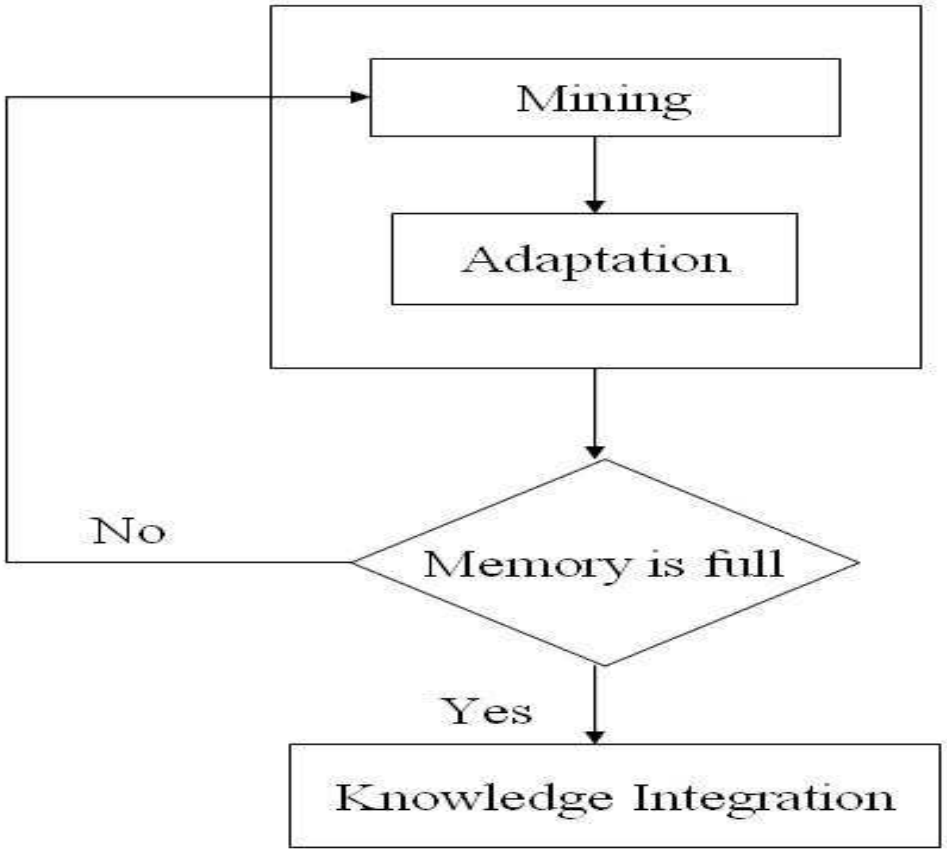

Figure 1. Algorithm Output Granularity

2.2.ANNCAD Algorithm

Law et al have proposed an incremental classification algorithm termed as Adaptive Nearest Neighbor Classification for Data-streams (ANNCAD). The algorithm uses Haar Wavelets Transformation for multi-resolution data representation. A grid-based representation at each level is used.

The process of classification starts with attempting to classify the data record according to the majority nearest neighbors at finer levels. If the finer levels are unable to differentiate between the classes with a pre-specified threshold, the coarser levels are used in a hierarchical way. To address the concept drift problem of the evolving data streams, an exponential fade factor is used to decrease the weight of old data in the classification process. Ensemble classifiers are used to overcome the errors of initial quantization of data.

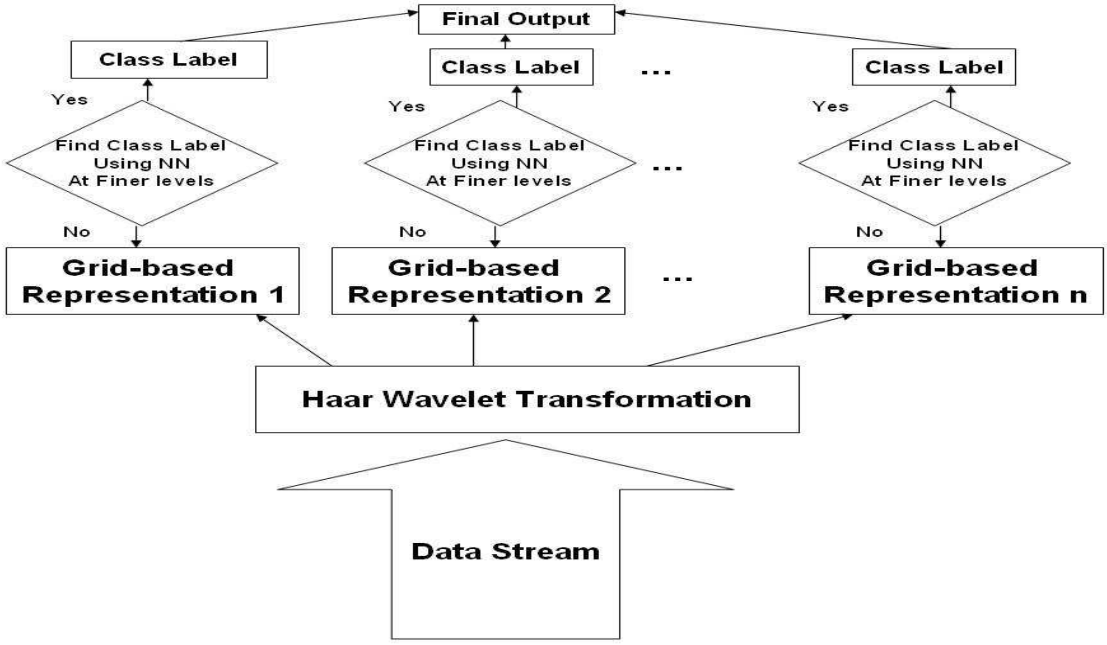

Figure 2. ANNCAD Framework

2.3. SCALLOP Algorithm

Ferrer-Troyano et al have proposed a scalable classification algorithm for numerical data streams. This is one of the few rule-based classifiers for data streams. It is inherently difficult to construct rule based classifiers for data streams, because of the difficulty in maintaining the underlying rule statistics. The algorithm has been termed as Scalable Classification Algorithm by Learning decision Patterns (SCALLOP).

The algorithm starts by reading a number of user-specified labeled records. A number of rules are created for each class from these records. Subsequently, the key issue is to effectively maintain the rule set after arrival of each new record.

- On the arrival of a new record, there are three cases:

1. **Positive covering:** This is the case of a new record that strengthens a current discovered rule.

2. **Possible expansion:** This is the case of a new record that is associated with at least one rule, but is not covered by any currently discovered rule.
3. **Negative covering:** This is the case of a new record that weakens a currently discovered rule.
   - For each of the above cases, a different procedure is used as follows:-
   1. **Positive covering:** The positive support and confidence of the existing rule is re-calculated.
   2. **Possible expansion:** In this case, the rule is extended if it satisfies two conditions:
      a. It is bounded within a user-specified growth bounds to avoid a possible wrong expansion of the rule.
      b. There is no intersection between the expanded rule and any already discovered rule associated with the same class label.
   3. **Negative covering:** In this case, the negative support and confidence is re-calculated. If the confidence is less than a minimum user-specified threshold, a new rule is added.

After reading a pre-defined number of records, the process of rule refining is performed. Rules in the same class and within a user-defined acceptable distance measure are merged. At the same time, care is taken to ensure that these rules do not intersect with rules associated with other class labels. The resulting hypercube of the merged rules should also be within certain growth bounds. The algorithm also has a refinement stage. This stage releases the uninteresting rules from the current model. In particular, the rules that have less than the minimum positive support are released. Furthermore, the rules that are not covered by at least one of the records of the last user-defined number of received records are released.

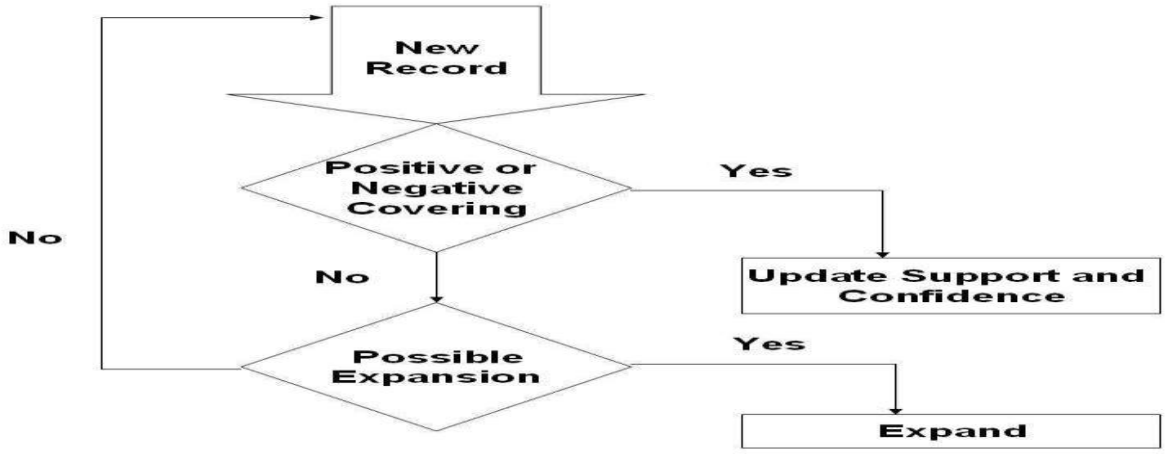

Figure 3. SCALLOP Process

3. Application areas in datamining, where it can be applied and Used

Data streams are continuous flows of data. Examples of data streams include network traffic, sensor data, call center records and so on. Their sheer volume and speed pose a great challenge for the data mining community to mine them.

Data streams demonstrate several unique properties:

- infinite length,
- concept-drift,
- concept-evolution,
- feature-evolution and
- limited labeled data.

Concept-drift: - occurs in data streams when the underlying concept of data changes over time.

Concept-evolution: - occurs when new classes evolve in streams.

Feature-evolution: - occurs when feature set varies with time in data streams.

Data streams also suffer from scarcity of labeled data since it is not possible to manually label all the data points in the stream. Each of these properties adds a challenge to data stream mining.

Multi-step methodologies and techniques, and multi-scan algorithms, suitable for knowledge discovery and data mining, cannot be readily applied to data streams. This is due to well-known limitations such as bounded memory, high speed data arrival, online/timely data processing, and need for one-pass techniques (i.e., forgotten raw data) issues etc.

In spite of the success and extensive studies of stream mining techniques, there is no single tutorial dedicated to a unified study of the new challenges introduced by evolving stream data like change detection, novelty detection, and feature evolution. This tutorial presents an organized picture on how to handle various data mining techniques in data streams: in particular, how to handle classification and clustering in evolving data streams by addressing these challenges.

The importance and significance of research in data stream mining has been manifested in most recent launch of large scale stream processing prototype in many important application areas. In the same time, commercialization of streams (e.g., IBM InfoSphere streams, etc.) brings new challenge and research opportunities to the Data Mining (DM) community. In this tutorial a number of applications of stream mining will be presented such as adaptive malicious code detection, on-line malicious URL detection, evolving insider threat detection and textual stream classification.

4. What can be done?

Data Stream Mining is the process of extracting knowledge structures from continuous, rapid data records. A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only once or a small number of times using limited computing and storage capabilities.

A process is a series of actions or steps repeated in a progression from a defined or recognized "start" to a defined or recognized "finish." The purpose of a process is to establish and maintain a commonly understood flow to allow a task to be completed as efficiently and consistently as possible.

Common business processes include purchase to pay (P2P), order to cash (O2C) and customer service. While nearly every company has some version of these processes as the backbone of their business, there are many others that support a company's daily operations:

- Manufacturing -process
- Distribution -process
- Logistics -process
- Supply Chain -process
- Accounts Payable -process
- IT -service management -process
- Accounts Payable
- IT Service Management
- Utilities
- Master Data Management

According to market research firm IDC, not only are most companies unaware of lost potential due to process weaknesses, 20-30 percent of their revenue is lost that way!

Think about all the processes, process steps and people involved in completing a task. Not only are there infinite possibilities for variables, different departments are responsible for different pieces along the way. That means there's rarely one person or team with oversight of all steps involved. It also means if one area is underperforming it impacts all the others, but it might not be immediately obvious where or how.

It's important to recognize that processes are not static. Even the best plans have exceptions, and over time these exceptions can become the rule. Dynamic markets also force change: customer expectations, new product lines, acquisitions, changing geographies, outsourcing, different suppliers, competitor moves, rules and regulations, etc.

When everything operates efficiently, a company is agile enough to adapt easily to outside forces, leaving more time to drive revenue through internal innovation, quality improvement and strengthening customer relationships.

Where can process mining help?

In our modern digital economy, companies require flexible processes to be competitive. That flexibility only comes through a deep understanding of how things are working and where shifts are possible.

- Which vendor gives you the best chance of meeting a committed delivery date?
- Which employees are not following the process?
- Which of your channel partners are down selling rather than upselling?

Process mining provides answers to these questions and more.

In addition, process mining allows you to quickly audit your processes, and many companies are using process mining for ongoing monitoring and optimization. That way they can detect potential problems before they have a negative impact, ensuring business operations are cost effective, compliant and several steps ahead of the competition.

5. Conclusion

Learning from data streams is an increasing research area with challenging applications and contributions from fields like data bases, learning theory, machine learning, and data mining. Sensor networks, scientific data, monitoring processes, web analysis, traffic logs, are examples of real-world applications were stream algorithms have been successful applied. Continuously learning, forgetting, self-adaptation, and self-reaction are main characteristics of any intelligent system. They are characteristic properties of stream learning algorithm

It is introduced Data Streams analysis. In my opinion, the usage of these techniques will be requested and more demanded in the following years due to the huge data produced every day. In my point of view, Stream Mining helps to analyses huge quantity of data solving two challenges: the impossibility to store all incoming data and to give results in nearby real time.