EDUZONE: International Peer Reviewed/Refereed Multidisciplinary Journal (EIPRMJ), ISSN: 2319-5045
Volume 11, Issue 1, January-June, 2022, Impact Factor: 7.687 Available online at: www.eduzonejournal.com

Framework to find Representative Instances in Data Streams Using Random Sampling

Abdul Razak M S[1], Dr. Nirmala C R[2]

[1,2]CS & E Department, VTU University, Karnataka – India

ABSTRACT

There is a lot of data explosion in today's dynamic environment. Many applications are continuously generating data and require an immediate response. The term "data stream" refers to a continuous stream of data arriving throughout time. Mining data streams is a challenging task due to its characteristics like size, speed and diverse data arriving over the period of time. This paper addresses the characteristic volume of data that cannot be stored entirely in main memory for processing due to lack in computation facilities. We use a probabilistic technique simple random sampling without replacement to extract the samples from the given population of data over a period of time. An incremental classifier very fast decision tree is used to classify the data streams for the extracted sample. Accuracy, response time, memory consumption of the samples are recorded and compared with the population data.

Keywords: Data streams, Random Sampling, Very fast decision tree

1. INTRODUCTION

Long-established data mining and machine learning algorithms usually work directly on the data. Batch algorithms will consume the entire data at each step, whereas a portion of the data will be consumed at each step by incremental and online algorithms (for example, one instance, successive instances, or a random sample). The later methods are becoming increasingly important for applications because of rising data volumes and a trend toward data streams (e.g., click data, sensor data, tweets, etc.). As a result, many traditional chores must now be completed online. As machine learning matures, researchers and practitioners begin to think about more difficult application scenarios. Learning from data streams is one example.Where data arrives in a continuous and non-stationary fashion. The field of machine learning has recently been quite interested in data streams. As a result, we have seen a growing number of algorithms for various applications, including classification, clustering, and anomaly detection. [1-4]

In many firms, extracting information from data streams is critical for inferring business patterns and, as a result, focusing resources on the correct markets and taking advantage of business possibilities. For example, in stock exchanges, investors must make quick decisions based on stock exchange data streams. Delays in making choices about new entrants could have serious effects.

Other issues develop when memory space is insufficient for building analytics over data streams, in addition to time limits. The solution is to use random sampling without replacement to extract the samples for the given period of data. This article usesrandom sampling technique to resolve time limit and memory for processing the data samples. The extracted samples will be used as an input to an incrementalclassifier to measure the accuracy, memory consumption and response time.

Data streams refer to a continuous stream of data arriving throughout time. The characteristics of data streams are unbound in size i.e. infinite length; data arrives in rapid i.e. speed arrival of data and nature of data changes [1-4].
Weuse the following techniques to discover the representative instances in data streams.

- Simple statistics from data streams are kept. (Parameters like mean, variance, standarddeviation, correlation)
- Sliding windows (sequence-based window and Timestamp based window)
- Count-based window: The number of observations in the sequence determines the size of the window.
- Timestamp-based window: The duration of the window determines its size.

A Classic Example for the sliding window for data streams is ADWIN (i.e. Adaptive Window)

- Data reduction (To overcome the computational challenges like processing the data element must be in less time and using a limited amount of memory.)
- **Data reduction techniques are:** Sampling, synopsis, histograms and wavelets, dimensionality reduction.

- **Example:** Stratified Sampling is a well-known method for maintaining an online randomsample..
- To approximate the frequency distribution of element values in a data stream, the summarizing techniques synopsis and histograms can be utilized.
- **Wavelets:** Wavelets are a numerical function that attempts to represent trends. Example: Haar wavelet transformation.
- **Dimensionality reduction:** This is a familiar preprocessing strategy to handle high-dimensional data, making any mining algorithm more expensive. It entails converting higher-level dimensional objects to a lower-level dimensional representation.
- **Discrete Fourier Transform:** It is used to derive a frequency-domain (spectral) representation of the signal.
- **Clustering Techniques:** Micro-clusters is a stream clustering method for storing summary information about the instances in the and the clusters.

BACKGROUND

Sampling

A sampling algorithm is a procedure for randomly selecting a subset of units (a sample) from a population., rather than having to count all possible samples.[18]

Let U = {1,...,k,...,N } represent a finite population, and s $\subset$ U denote a sample or subset of U. A probability distribution on the set of all subsets is a sampling design p(s). i.e s $\subset$ U, i.e p(s)$\geq$ 0 and

$$\sum_{s \subset U} p(s) = 1$$

Simple random Sampling, Stratified Sampling, and cluster sampling are examples of various sampling procedures. Random Sampling is a simple method. In a simple random sample, each possible sample of size n from a population of N items has an equal probability of being chosen. (Simple Random Sample). It can be picked at random from a range of numbers between 1 and N. Simple random sampling ensures that all things have an equal chance of being chosen, giving it an equal probability approach to selection.

The population data is divided into smaller groups in Stratified Sampling, and a random sample is obtained from each group.

In cluster sampling Instead of selecting from each group, the entire population is divided into smaller groups and then the entire group is randomly selected.

Very Fast Decision Tree / Hoeffding Tree

A Hoeffding Tree [19] is a decision tree induction system that may incrementally and at any time learn from large-scale data streams, as long as the distribution samples do not change over time.. Hoeffding trees take advantage of selecting the best splitting attribute usually requires only a small sample size.. The Hoeffding bound mathematically supports this notion, which determines the number of observations required to estimate given statistics within a specified precision (in our case, the usefulness of an attribute). Hoeffding trees have a conceptually desirable characteristic that other incremental decision tree learners lack i.e. Good performance guarantees.The Hoeffding bound can be used to show that an incremental learner with an unlimited number of instances produces output that is asymptotically substantially equivalent to that of a non-incremental learner.

The Hoeffding bound is defined as:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta))}{2n}}$$

ϵ: Hoeffding bound.

R: A random variable's range. The probability range is 1, and the range of an information gain is log c, where c is the number of classes.

δ: Confidence. 1 minus the required probability of selecting the correct property at each node.

n: sample count.

LITERATURE SURVEY

Madden et al. Arasu et al. [5,6] defined Sampling over data streams as an approach for quickly inferring information without knowing the length of the stream, which is particularly useful for continuous inquiry systems.

Gama et al [7] The Reservoir Sampling algorithm is a standard and commonly used algorithm for sampling data streams. However, it necessitates a fixed-size reservoir in techniques based on it [8,9].

Gemulla and Lehner,Babcock, Braverman et al.[10,11,12] use moving/sliding windows over data streams where the timeline is important.These methods assume that old data is no longer valid in a time interval t (timestamp-based sample) or a group of the n most recent components (sequence-based sample). The most recent components in a window are replaced to form a sample of size k. They do, however, necessitate the definition of a fixed-size sample (k) and additional information (e.g., t or n), resulting in the use of O(k) memory space and O(k log n) for bursty windows.

Manku and Motwani ,Gibbons [13,14] concentrate on the frequency of items in order to sample things from the stream at any time. For example, Hybrid-Streaming [15] considers the frequency of items across numerous data streams while preserving approximate histograms for all data streams from various report sources. The algorithms collect frequency summaries and store them in internal structures, which takes up more memory.

Levin and Kanza [16] developed a distributed MapReduce algorithm. This method seeks to build multi-survey stratified Sampling over online social networks while taking into account unique limits and costs associated with sharing individuals between surveys. The strata formed by the Map and Reduce functions are used to pick elements. On the other hand the algorithm, necessitates the population size from which the stratum is extracted.

Al-Kateb et al. [17] provided a dynamic reservoir size modification while Sampling was still in progress to deal with the static characteristic of reservoir sampling. The reservoir size and sample homogeneity were the two key issues they focused on. The reservoir size is modified based on runtime data properties or application behavior.

METHODOLOGY

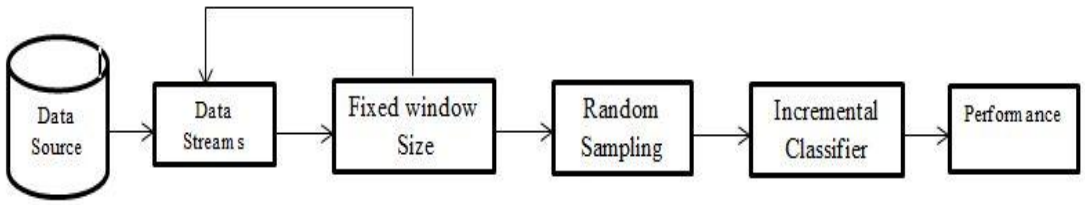

Figure 1: Sampling Process from the data streams

The data from the source will be read continuously and stored in a sliding window of size n for processing. Once the data in the window is available, we apply the random Sampling without replacement technique and extract the sample size from the window data and in turn the sample size is given to an incremental classifier VFDT(very fast decision tree) for classification.This procedure is repeated until there is no more data in the data source.

Algorithm: Stratified Sampling

Input: Dataset

Output: Sample of the dataset

Initialization: Read data continuously in a stream fashion

do {

Read the stream data into the defined window size

Apply random sampling without replacement to the window data

Sample data are given to the incremental classifier

Measure the performance for sample data

} **while**(no more records exist in the data source)

Experimental setup

The experiment is demonstrated on a 64-bit machine with 8GB RAM and Windows 10 operating system. The program is implemented in python language using thescikit-multiflow python framework for data streams.

RESULTS AND DISCUSSION

Datasets Description
In our experiment we have considered the datasets from the real and synthetic world like spam, weather, bank_loan, electrical, dermatology, transient chessboard. The number of records and features and class labels in our datasets are briefed in the below table

Table 1: Datasets Details

Dataset name	# of records	# of features	# of class labels
spam	6214	499	02
electrical	45313	06	02
bank_loan	5000	13	02
Dermatology	367	34	06
Transient chessboard	200001	02	02
weather	18160	08	02

The data in the Table 1 is numeric and real, andthe target variable is binary and multiclass form.

Table 2: Population data description

Dataset	# of Records	# of Features	Time Taken (in seconds)	Memory Consumption (in bytes)	Accuracy
Spam	6214	499	73.923	2933804	86.224
electrical	45313	06	5.859	188204	77.461
bank_loan	5000	13	1.011	68164	73.380
Dermatology	367	34	0.393	114474	74.659
transient Chessboard	200001	02	31.910	508684	61.30
weather	18160	08	3.285	69644	50.567

Table 2 describes the population data. It describes the details about the dataset such as count of records, count of features and time taken (in seconds), memory consumption (in bytes) for processing the entire population. The VFDT classifier is applied to the population data to measure the accuracy.

Table 3: Sample data description

Dataset	Sliding Window Size / strata	Sample Size	# of Samples Analyzed	Time Taken (in seconds)	Memory Consumption (in bytes)	Accuracy
Spam	1000	100	700	1.431	792090	78.428
electrical	1000	100	4600	0.011	36212	76.282
bank_loan	1000	100	500	0.023	47304	70
Dermatology	100	50	200	0.149	114090	77.5
transient Chessboard	10000	5000	100000	0.953	203092	41.308
weather	1000	100	1900	0.014	16894	67.842

Above table 3 describes the sample data. It describes the details about the dataset such as window size , samples drawn from the window, the total number of samples analyzed, time taken (in seconds), and memory consumption (in bytes) for processing the sample data. The VFDT classifier is applied to the sample data to measure the accuracy.

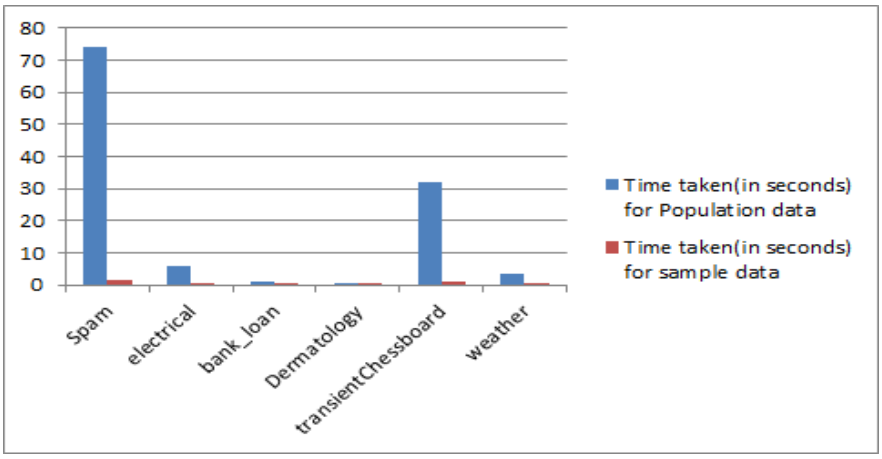

Figure 2: Time spent in comparison (in seconds) for population and sample data

The above bar graph in figure 2 displays the time spent comparing (in seconds)for population and sample data for different datasets. The X-axis represents different datasets, and the y-axis represents the time taken in seconds.

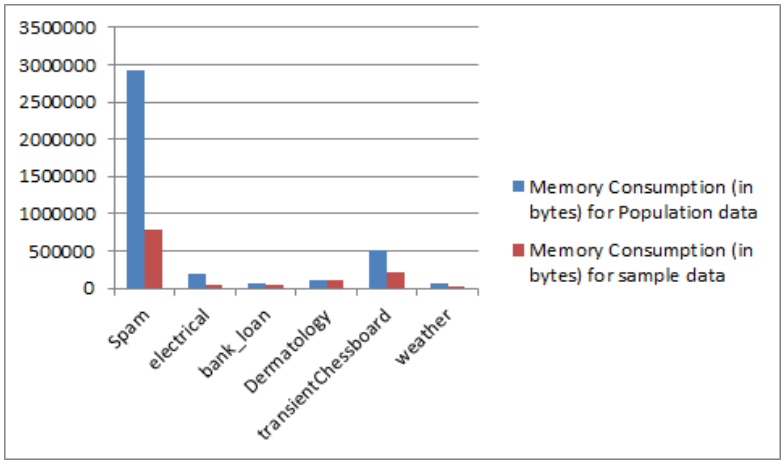

Figure 3: Comparison of memory consumption(in bytes) for population and sample data

The above bar graph in figure 3 displays the comparison of memory consumption (in bytes) for population and sample data for different datasets. The X-axis represents the various datasets and the y-axis represents the memory consumption in bytes.

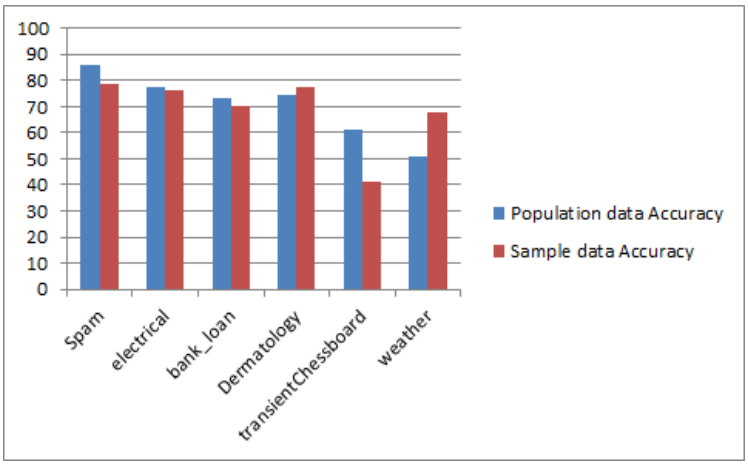

Figure 4: Comparison of accuracy for population and sample data

The above bar graph in figure 4 compares accuracy for population and sample data for different datasets. In the above figure 4 the x-axis represents the datasets and y-axis represents the percentage accuracy.

CONCLUSION

This paper aims to note the representative instances for data streams through arandom sampling technique. In this paper, we have considered the parameters like time taken, memory, and accuracy for the sample data and population data. The dataset we have considered is numeric in nature. Different techniques for finding the representative instances can be considered and compared with our results in the future and also in future categorical datasets can also be considered and incremental ensemble classifiers can also be used to find the accuracy of sample data rather than considering a single classifier to find accuracy.

REFERENCES

[1] Hill DJ, Minsker BS (2010) Anomaly detection in streaming environmental sensor data: a data-driven modeling approach. Environ Model Softw 25(9):1014–1022.
[2] Dyer KB, Capo R, Polikar R (2014) Compose: a semi supervised learning framework for initially labeled nonstationary streaming data. IEEE Trans Neural Netw Learn Syst 25(1):12–26.https://doi.org/10.1109/TNNLS.2013.2277712.
[3] Fanaee-T H, Gama J (2014) Event labeling combining ensemble detectors and background knowledge. ProgArtifIntell 2(2):113–127.https://doi.org/10.1007/s13748-013-0040-3.
[4] Nguyen HL, Woon YK, Ng WK (2015) A survey on data stream clustering and classification. KAIS 45:535–569.https://doi.org/10.1007/s10115-014-0808-1.
[5] Madden, S. and Franklin, M. J. Fjording the Stream: an architecture for queries over streaming sensor data. In Proceedings of the 18th International Conference on Data Engineering, ICDE. San Jose, CA, USA, pp. 555–566, 2002
[6] Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., and Widom, J. STREAM: the stanford stream data manager (demonstration description). In Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, NY, USA, pp. 665–665, 2003.
[7] Gama, J. Knowledge Discovery from Data Streams. CRC Press, Boca Raton, USA, 2010
[8] Chen, C., Yin, H., Yao, J., and Cui, B. TeRec: a temporal recommender system over tweet stream. Proceedings of the VLDB Endowment 6 (12): 1254–1257, 2013.
[9] Bonin, R., Marcacini, R. M., and Rezende, S. O. Unsupervised Instance Selection from Text Streams. Journal of Information and Data Management 5 (1): 114–123, 2014.
[10] Gemulla, R. and Lehner, W. Sampling Time-based Sliding Windows in Bounded Space. In Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, NY, USA, pp. 379–392, 2008.
[11] Babcock, B., Datar, M., and Motwani, R. Sampling from a Moving Window over Streaming Data. In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA, USA, pp. 633–634, 2002.
[12] Braverman, V., Ostrovsky, R., and Zaniolo, C. Optimal Sampling from Sliding Windows. In Proceedings of the

ACM Symposium on Principles of Database Systems. New York, NY, USA, pp. 147–156, 2009

[13] Manku, G. S. and Motwani, R. Approximate Frequency Counts over Data Streams. In Proceedings of the International Conference on Very Large Data Bases. Hong Kong, China, pp. 346–357, 2002.

[14] Gibbons, P. B. Distinct Sampling for Highly-Accurate Answers to Distinct Values Queries and Event Reports. In Proceedings of the International Conference on Very Large Data Bases. San Francisco, CA, USA, pp. 541–550, 2001.

[15] Guo, J., Zhang, P., Tan, J., and Guo, L. Mining Frequent Patterns Across Multiple Data Streams. In Proceedings of the ACM International Conference on Information and Knowledge Management. New York, NY, USA, pp. 2325– 2328, 2011.

[16] Levin, R. and Kanza, Y. Stratified-sampling over Social Networks Using Mapreduce. In Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, NY, USA, pp. 863–874, 2014

[17] Al-Kateb, M., Lee, B. S., and Wang, X. S. Adaptive-Size Reservoir Sampling over Data Streams. In Proceedings of the International Conference on Scientific and Statistical Databases Management. Washington, DC, USA, pp. 22–34, 2007.

[18] MiodragLovric "Sampling Algorithms" , International Encyclopedia of statistical science" 2011.

[19] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In KDD'01, pages 97–106, San Francisco, CA, 2001. ACM Press.