IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

Optimal Strategy for Monitoring Top-k Queries on Document Streams

V.Bhramaramba,PG Scholar, Dept. of MCA, ,VVIT COLLEGE,NAMBUR

P.Pothuraju ,Associate Prof, Dept. of CSE, P.Pothuraju,VVIT College, NAMBUR.

Abstract: In the present days, the accumulation of the information is of high that leads to the innovation of novel field, named, Big Data. Most of the online monitoring applications that exhibits stream of data such as call records, sensor readings, web usage logs, and network packet traces etc are to be preprocessed effectively for future aspects. Though, data has been emerged from different sectors, it has to be interpreted and stored in an effective way. A prior scheme suggests frequency and indexing order based data arrangements process that explores complex task in the view of big data systems. This paper focuses on designing a novel data arrangements and interpretation model for the data streams. We propose a tokenized top k query handling model which reduces the recomputational and memory utilization. Each data object is represented by frequency, identity and the lifespan of the objects. Relied upon the lifespan of the objects, its subsets are predicted and then formulated to the top k process. These top k objects are continuously analyzed and then declared as the 'tokenized top k set'. By doing so, we can efficiently achieve the less memory consumption with an effective preprocessed data objects. Experimental analysis have been studied in the synthetic IBM T10I4D100K dataset in the terms of no. of generated candidate sets, time taken for first scan, time taken for second scan and the memory usage. It is compared with the existing, frequency ordering approach which proves that our proposed model achieves better results in memory usage and recomputational tasks.

Keywords: Big data, Data streams, top k queries, Objects, Frequency and the Memory utilization

1. Introduction

Nowadays, a tremendous volume of data is being generated using variants technologies. In accord to the Digital Information System (DIS), the information growth will be 20 times by 2020 [6]. The discovery of

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

knowledge from those data sorts is highly challenging tasks in the stream of document. In order to derive relevant knowledge from the pool of data, data extraction techniques are widely suggested. Data mining is the recent and novel mechanism [5] in the knowledge discovery process. In specific to, document streams refers to data obtained from news, micro-blog articles, rapid messages, web forums, threads etc. It mainly focuses on the specific topics. The knowledge can be extracted using several domains like sequential mining, pattern mining, frequent mining etc.

The relevant knowledge from the obtained data plays a vital role in the data mining tasks [10]. The similar patterns have to be extracted from the group of data using the concepts such as association rules, sequences, correlations, classifiers etc. Data mining is the basic task of the big data systems. The major problem with frequent set mining methods presented previews is the explosion of the number of results; it is difficult to find the most interesting frequent item sets [11]. We are facing the following disadvantages: many transactions, huge database, many data and not enough information. Large sets of frequent item sets describe essentially the same set of transactions. This problem was approached in the paper „Item Sets That Compress", It uses the MDL principle to reduce the number of the item sets: the best set of frequent item sets is that set that compresses the database set. The problem of frequent item set mining was extended to sequential pattern mining. By the issue of finding frequent sequences we are facing with the problem of having a big number of frequent sequences and many redundancies. The article "Mining conjunctive sequential patterns" presents an algorithm for non-derivable conjunctive sequential patterns [14] and shows its use in mining association rules for sequences.

To limit the output size and to control the itemsets with the highest utilities without setting the thresholds, a better solution is to change the task of mining HUIs as mining top-k high utility itemsets [12]. Here the users specify k. Here k is the number of desired itemsets, instead of specifying the minimum utility threshold. Setting k is

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

easier than setting the threshold because k is the number of itemsets that the users want to find whereas selecting the threshold depends on database characteristics, which are unknown to users.

2. Related Work

This section describes the prior works on the document streams. In the perspective of the data streams [4], a different attributes and its properties plays a vital role in the data classification process. In order to store and retrieve the stream of objects incessantly, an algorithm should be designed effectively. The author presented a top k frequent items set model for the data stream using sliding window concepts. The idea behind this is to extract the topmost items which preserve the memory usage [3]. But it shows higher consumption of the memory usage. The similar study was extended by author that intelligently classifies the incoming objects based on the training and testing data streams model. They created a classification model that dynamically classifies the incoming objects. Their model classified the incoming objects using training features.

A hash based data stream approach was introduced [12]. The incoming objects are computed into the hash tables using linear congruencies. They were also studied about the variant of data streams such as offline and online data streams. The most frequently used patterns are labeledinto a class. The author depicted a singular value decomposition model for handling multiple data streams. It was specifically designed for the offline data streams systems. In specific to, they discussed about the clustering concept in the text data stream. It was discussed in the spam detection, unusual words removal etc. The author studied about the clustering data streams using the semantic modeling systems. It was also solved clustering algorithms. Their model was applied to the instant message applications and the internet relay chat.

Generally, a tremendous volume of data has been generated by the social networks systems. The detection of threads in social data system is a challenging task [13]. They have studied about the single pass clustering algorithm using linguistic features. Dimensionality is the main issue in the data

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

stream. Conventional learning vector quantization was suggested as incremental learning model. A learning map was used to derive the relevant knowledge. This model helped to achieve the accuracy of the data. The author studied about the Hoeffding tree algorithm that makes use of information gain as metric. It required only a certain number of samples to conduct attribute split at a node. The number of split is calculated with the Hoeffding constraint equation [15]. To increase the exponential order of the algorithm accuracy, it is only needed to linearly increase the number of sample data, and the samples shall only be scanned for once for the algorithm.

The author discussed about the sliding window technology that minimized the time taken for data extraction process [11]. If there are concept drifts, an alternative attribute with higher information gain will appear and make some previous nodes which no longer meet the Hoeffding constraints. Then, the replacing subtree with better split attributes as root nodes begins to grow. The new growing sub-tree will replace the old subtree when its accuracy surpasses that of the old sub-tree. The author discussed about the concept drift mutation model which clustered the incoming data with some threshold limits. Yet, it consumed higher amount of memory usage. This method is supposed to analyze different concept drift types and propose detection method of concept drift with targeted characteristics.

3. Proposed Methodology

This section depicts the working model of our proposed work. We have designed a "tokenized top-k queries handling" which extracts the relevant and accurate data from the set of resources. Prior schemes make use of the top k analysis which throws computational overheads on the query window. It also consumed heavy memory since the data objects are live in nature. The proposed model is explained as follows:

Formation of the database: This is the first step of the proposed model which forms the database of the systems. Generally, it's a cumbersome task to label the stream of data. Thus, we have formulated the problem in a systematic way. Let D be the database with the top k query Q (D, A, n) where n is the

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

data objects and A is the preference function that holds the top k list in present time. In some cases, it holds one or more attributes of complex objects. In our system, a time frame is inserted into each query which will be active for certain period of time. Let the objects O be O1…O2…On where n is the stream of objects. The incessant arrival of data objects are queried as Q {Frame F, Object Interval (Obji), n} which works on each object. It executes on the liveliness of the data frames.

Minimal top k candidate set: This steps help to achieve the accurate candidate set of the obtained objects. Initially, each data objects are itemized in an order with its basic class label. Utility list is maintained for the data objects. Each data frame is verified twice for grouping into its relevant candidate sets. In the first scan, the object ID and time are verified and then ordered. And the second scan verifies the preference function A with its stipulated period of time. The

A:→{Item Id, Object Id, Object time}

The stream of data objects that satisfies the preference function will group into minimal candidate sets.

Tokenized Top k sets: This steps help to achieve a tokenized top k with the incremental administration process. Each data object operates on the time window. Relied upon the incoming objects, the time frame is renewed. Once the object time gets expired, then the data is not encountered into the top k sets. Initially, an empty set is created that holds the tokenized top k sets. Relied upon the time of the objects, the incoming objects are handled. Tree based data structure is used to store and retrieve the objects. No matter which data structure is chosen, the best possible CPU costs for inserting a new object into a top-k object set and keeping the size of the top-k object set unchanged has complexity O(log (k)). More precisely, log (k) for positioning the new object in the top-k object set, and log (k) for removing the previous top-k object with the lowest A score. Thus, the overall processing costs for handling all new objects for each window slide is O(Nnew ∗ Cnw_topk ∗ log (k)), with Nnew the number of new objects

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

coming to the system at that time, and Cave_topk is the average number of windows of each object to achieve tokenized top k, when it arrives at the system. As the object expiration process is trivial, this constitutes the total cost for updating the top-k result at each time. The fig.1 presents the formation of top k results.

Extracting the tokenized top k in single phase: The tokenized top k results holds two parts, namely, arrival of incoming objects and the computation of the preference object in the query window. Only the preference objects are computed for the tokenized top k results. The knowledge patterns are derived for each incoming objects. By doing so, we have achieved the efficiency of the system. Since, the database is scanned twice, the searching complexity is also reduced.

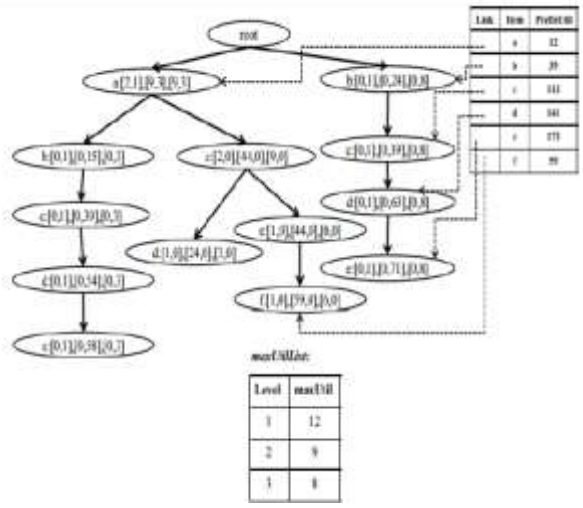

Figure 1 Formation of top k in tree access structure

4. Experimental Results

This section depicts the experimental analysis of our proposed model. The experimental analysis is to find the tokenized top k objects over the data stream. Synthetic IBM dataset T10I4D100K is collected which contain average transaction size T, average frequent patterns I and the number of transactions D. Thus, the dataset contain 100,000 transactions with 870 frequent patterns of average length 10.1. It makes batch size of 10,000 with 5 query windows. With the synthetic dataset, we analyzed the performance of the following.

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018

ISSN: 2320-1363

No. of generated candidate set: This metric assist to discover the effectiveness of the generated candidate sets. The table 1 depicts the no. of generated candidates set between proposed tokenized top k and existing frequency ordered indexing approach.

Table 1 No. of generated candidates set based on the threshold k.

Dataset	K	Proposed	Existing
IBM	100	2852017	69986
	200	4985632	84895
	300	8177112	94850
	400	10118345	100847

Time taken for first scan: The time taken for first scan depicts the total execution time taken for indexing the incoming objects. It is measured in the seconds (Sec). The fig.2 depicts the comparison graph between proposed and existing model. It is inferred that the proposed model achieves better time complexity than the existing model with lessened k values.

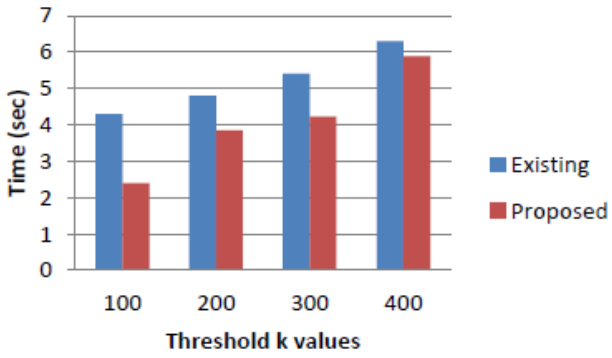

Figure 2 Time taken for the first scan of the incoming objects

Time taken for second scan: The time taken for second scan depicts the total execution time for the preference objects generation. It is also measured in sec. The fig.3 shows the comparison graph between proposed and existing model. The existing model makes use of three scan to index the incoming objects. Yet, the proposed model make use of twice scan to incessantly index the incoming objects and then generating the tokenized top k sets. To the best of our knowledge, within twice scan of the database, the proposed model achieves better classification accuracy.

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018

ISSN: 2320-1363

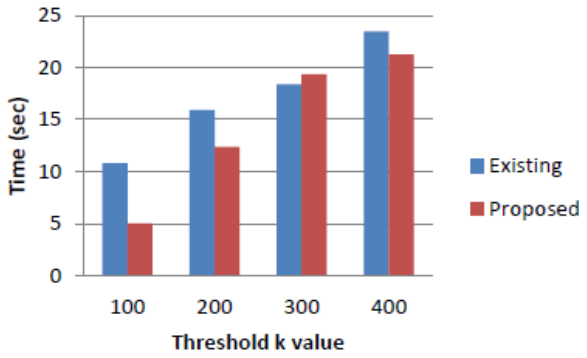

Figure 3 Time taken for second scan

Memory consumption: Memory consumption is the major performance metric studied for constructing tree based data structure. The existing model consumes higher memory consumption in terms of database formation, identifier and frequency order of the frequent candidate sets. Whereas, the proposed model consumes lesser memory usage because all the incoming objects are assessed using preference functions and then ordered within stipulated period of time. The fig.4 depicts the memory usage between proposed and existing model.

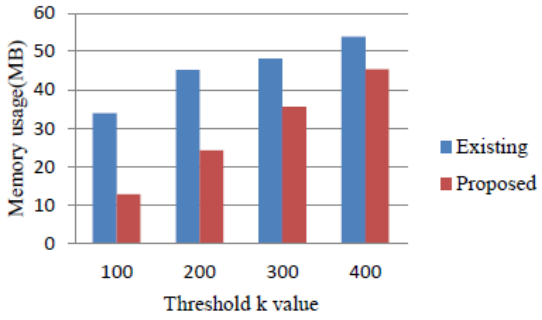

Figure 4 Memory usage

5. Conclusion

This paper focuses on designing a novel and efficient, „tokenized top k itemsets" that extracts the relevant and topmost frequently used data in query windows over the stream of data. The objective of this work is to prune the search space, recomputational and the memory usage for the classification of document stream. The incoming objects are scanned twice and then stored in the database for easy retrieval systems. The received objects are formulated into the query window which composes of the identity, frequency and time of the objects. Based on the liveliness of the data objects, it is moved onto the second scan. In this, preference function is computed for lived objects. The frequent patterns of objects are analyzed and then placed onto the utility list. This list composes of the preference objects

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018 ISSN: 2320-1363

with its accurate class label. By setting the threshold k value, the data objects which hold high utility rank are classified. Experimental analysis have been processed in synthetic IBM T10I4D100K dataset and the metrics studied are the no. of generated candidate sets, time taken for first scan, time taken for second scan and the memory usage. We have achieved our study motives by comparing the proposed results with the existing model which proves that proposed model consumed lesser memory usage and recomputational tasks.

6. References

[1] P. Merlin, A. Sorjamaa, B. Maillet, and A. Lendasse. X-SOM and L-SOM: A double classification approach for missing value imputation. Neurocomputing, 73(7-9):1103–1108, 2010.

[2] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In VLDB, pages 697–708, 2005.

[3] Papadimitriou, J. Sun, C. Faloutsos, and P. S. Yu. Dimensionality reduction and filtering on time series sensor streams. In Managing and Mining Sensor Data, pages 103–141. 2013.

[4] J. Sun, S. Papadimitriou, and C. Faloutsos. Online latent variable detection in sensor networks. In ICDE, pages 1126–1127, 2005.

[5] O. G. Troyanskaya, M. N. Cantor, G. Sherlock, P. O. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6), 2001.

[6] B. Yi, N. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris. Online data mining for co-evolving time sequences. In ICDE, pages 13–22, 2000.

[7] C. Yozgatligil, S. Aslan, C. Iyigun, and I. Batmaz. Comparison of missing value imputation methods in time series: the case of turkish meteorological data. Theoretical and Applied Climatology, 112(1-2), 2013.

[8] Leong Hou U et al, Continuous Top-k monitoring on Document Streams, IEEE transactions on knowledge and data engineering, 29(5), 2017.

[9] E. Keogh, J. Lin, and A. Fu. HOT SAX: Efficiently finding the most unusual time

series subsequence. In ICDM, pages 226–233, 2005.

[10] E. J. Keogh. Exact indexing of dynamic time warping. In VLDB, 2002.

[11] E. J. Keogh and T. Rakthanmanon. Fast shapelets: A scalable algorithm for discovering time series shapelets. In ICDM, pages 668–676, 2013.

[12] M. Khayati and M. H. Böhlen. REBOM: recovery of blocks of missing values in time series. In COMAD, pages 44–55, 2012.

[13] M. Khayati, M. H. Böhlen, and J. Gamper. Memory-efficient centroid decomposition for long time series. In ICDE, pages 100–111, 2014.

[14] M. Khayati, P. Cudré-Mauroux, and M. H. Böhlen. Using lowly correlated time series to recover missing values in time series: a comparison between SVD and CD. In SSTD, pages 237–254, 2015.

[15] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos. DynaMMo: Mining and summarization of coevolving sequences with missing values. In KDD, pages 507–516, 2009.

About Authors:

V.Bhramaramba is currently pursuing hisMCA in MCADepartment,VVIT College, NAMBUR,A.P. She received her Bachelor of degree(BCA) in computer Department from JKC College ANU university

Mr. P. Pothuraju is currently working as an Associate Professor inCSE Department, VVIT Engineering College, NAMBUR. He completed PhD in ANU University He completed M tech in SRKR engineering

IJMTARC – VOLUME – V – ISSUE – 22, APR - JUNE, 2018

ISSN: 2320-1363

college in Andhra University.He has a vast teaching experience is more than10 years.