www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 09 September 2016 Page No.18066-18070

Function of big-data over data streams

S. Mahammed Gouse [1], E.Amarnath Goud [2]
[1] *Avanthi Institute of Engineering and Technology,*
Gunthapalli, RR district, Telangana State, India
E-mail: s.mahammedgouse@gmail.com
[2] *Avanthi's Scientific Technological & Research Academy,*
Gunthapalli, RR district, Telangana State, India
E-mail: amar4goud@gmail.com

Abstract: ***"Data streams" is defined as class of data produced over "text, audio and video" channel in uninterrupted form. The streams are of unlimited length and may consist of ordered or unordered data. With these features, it is hard to store and process data streams with simple and static strategies. The processing of data stream poses four key challenges to researchers. These are countless length, concept-development, and concept-flow and feature development. Unlimited-length is because the amount of data has no limits. Concept- flow is due to sluggish changes in the theory of stream. Concept-development occurs due to existence of unfamiliar classes in data. Feature-development is due to development new features and deterioration of old features. To carry out any analytics data streams, the translation to knowledgeable form is important. The researcher in past have anticipated different strategies, most of the research is focused on difficulty of unlimited -length and concept-flow. The research work offered in the paper describes a well-organized string based methodology to process "data streams" and manage the challenges of unlimited length, concept-development and concept-flow. Subject areas Data mining, Machine learning.***

Keywords: Data stream, Data mining, Concept-flow, Concept-development, Novel, Features.

1. INTRODUCTION

With the development of technology and use of internet of things [IoT], the quantity of data generated over machine communication channels is exponentially growing. Data mining is one of the stream of "Database technologies" deals in processing huge volume of structured and unstructured data. primarily, it was complex to store and process the data produced over the communication channel, but in the present situation researchers have developed methodologies to overcome the limitation. The data produced in text, audio, video format and is flowing from one network node to another, in un-interrupted manner is denoted as "Data stream". The major characteristics of streaming data are: continuity, dynamic nature and no defined format. Its features keep on changing frequently, which makes it complex to process. The four main challenges in processing streaming data are: unlimited-length, concept-development, concept-flow and feature-development.

(i) The data stream is generated at very high speed and is unlimited in size. It is unrealistic to store and process stream for training the systems.

(ii) Concept flow is said to be there when the essential concept of the streams changes with time domain. This change causes the original classes of data, to drift towards the new characteristics.

(iii) Concept development take place when new classes evolve in the data. For instance, concept development occurs when a new class of virus signature is detected or a new class of network attack is detected. Such development at run time are complex to manage by any system.

(iv) Feature progression is the lengthy process. The new feature start appearing in the stream due to concept flow and concept progression. The development of new feature fades away the existing features and over the period of time, considerable changes are observed in the system.

The static data classification methods cannot be used in processing data streams due to four challenges [1–3]. There is a need to offer proficient classification techniques, which are appropriate to handle the data stream challenges. The main challenges and results presented along with proposed resolution is explained below:

a. countless length problem: Incremental learning forms have been proposed by researchers in which hybrid batch incremental processing method is used. The method divides the data streams into portions of equivalent sizes, and instructions are provided to classification model to process the chunks.

b. Concept flow problem: This difficulty can be recognized by examining the changes occurring in the streaming data. The changes happening in the streaming data are variable and is handled by the data models which needs normal updates as per the changes in the streaming data. Most of researcher have provided incomplete solution, by setting up the classes of data. In the following stage of research illustrated in the paper, the variability is formed by varying the number of classes of data, which enables the system to process innovative class of data.

c. Concept development problem: In the presented research work, the concept development problem is addressed by allowing the classifier to mechanically identify the novel class, without having any previous training about the novel class. In past, conventional classifiers, were not able to discover novel classes, unless trained.

d. Feature development problem: This obscurity is address in the present research work by continually monitoring of new features in the streaming data. The proposed result is based on string comparison operation. The model design is association of various models, in which, outliers from each model is detected and final outlier is found. The resultant outlier is used to separate instances based upon their occurrences, i.e. concept-development, concept flow or noise. The new feature is used to update the present model, to make possible it to handle the challenges of streaming data.

2. ASSOCIATED WORK

In most of the research work in past, researcher have efficiently solved the unlimited length and concept flow problem, but no main findings are reported for concept development and feature development problems. For solving unlimited length and concept flow problem, research have proposed various incremental approaches. These approaches are: Single

model incremental approach [4] and hybrid batch incremental approach [5, 6]. In incremental approach, only one model is used for categorization, which is dynamically updated at regular time interval. In Hybrid batch incremental approach, is collections of different models and batch learning techniques. In this approach, the model is produced from recent data and based on the efficiency of classification, it is discarded. The basic methodology of hybrid model makes it simple to implement and update An outlier is an examination of an unusual distance measured between a data value and all other values of data in a arbitrary data sample. The complexity of concept development and feature development can be solved by finding out outliers from the data samples. The outlier occurs in streaming data due to causes like: noise, concept development or concept flow. The presented research work intends at finding the causes of occurrences of outliers. This will keep away from misclassification of concept flow as outlier and reduce false alarm rate [misclassifying an existing class instance as novel class instance].The research work carried out by Spinosa et al. [7], presented a technique to handle concept development and concept flow, along with unlimited length problem. In [7], methodology described uses clustering technique to detect novel classes. The clustering is executed on normal data contained in the area specified by hyper sphere. The model is always updated as the stream progress. When the cluster is formed and if it lies exterior to hyper sphere, its density is verified and if it high, it is declared as novel class. This approach cannot be used for data having multiple classes, as it assumes only one class as normal class and rest of classes as novel class. The approach is classified as

one class classifier and it assumes that shape of occurrences of normal class in feature space is convex, although it is not so in practical. In the following research work carried out by researchers [2, 3, 8] the novel class detection strategies are classified as parametric and non-parametric. The parametric approach relates the normal range of data with distributed range to calculate distribution parameters. If an instance is not in-line with distribution parameter, it is classified as novel class. The non-parametric methods are not based on data distribution and therefore not restricted. The research work presented in paper is based on non-parametric approach. Also majority of approaches presented in the research work by researcher [1–3] can discover presence of only one novel class. The research work presented in the paper explains an approach to identify multiple novel classes and is categorized as multiclass classifier.

3. SUGGESTED APPROACH

In the presented research effort, the classifier functions on collection of three models. In the proposed approach the data stream will be either classified into an existing class or into a novel class. Let "L" represents a group of models {M_1, M_2, M_3,… M_n}. Following definitions are used in the proposed approach.

Description 1 *Existing class* If a model M_i that belongs to a group is trained by a class 'C' and defines it, then class 'C' is called an existing classes. In other words at least one model belonging to group M must be trained on class C.

Description 2 *Novel class* If class 'N' is not known to any of the models M_i belonging to group M, then 'N' is a novel class. No model of the group has been trained on novel class.

Description 3 *Outliers* If x is a test instance and if doesn't match the specifications of

any of the class 'C' of the model M_i then 'x' is an outlier of the model M_i. The outliers don't belong to any of the class defined by the model.

4. TRAINING PHASE

In the training phase the training data is separate into equal sized chunks. For experimentation and smooth handling the size of each chunks is set to 2000 tuples. The division generates different number of classes in each chunk. These classes are calculated by applying K-medoid clustering technique on each chunk. The K-medoid technique performance is more suited to data set holding outliers [9]. The training phase will produce separate model for each chunk of data on which training is performed. The model is stored as number of clusters created and set of words (S_i) defining the cluster. The classification rule followed by group is: If 'X' is an occurrence to be tested, it is submitted to each model M_i in the group to check whether it is an outlier for model M_i. If it is not an outlier (OUT), it will be classified by model M_i into one of its classes and if it is identified as an outlier by all the three models then it will be considered as a final outlier i.e. (FOUT).

5. OUTLIER DETECTION

The training phase generates three models, which are stored in the form of number of clusters and set of words defining the clusters. The models are used to detect the outliers for test instances. The test instance is submitted to each model M_i for classification.

Step 1 To classify the test instance, the words present in the test instance are collected. The check is carried out to determine if these words are present in the set of words S_i defining any class "C" of the model. If test instance words are present in the set S_j of class 'C_j' then the instance is classified as belonging to the class 'C_j' of model 'M_i'. If the test instance does not belong to any of the class defined by the model 'M_i', it is declared as an outlier (OUT) for that model 'M_i'.

Step 2 This step will find the final outliers of the group. The outlier detected in step 1 for each model M_i are stored in separate vector 'OUT_i'. Each "OUT_i" is checked to find out a common instance present in all outlier arrays. If such instance is found, it is declared as "FOUT" and all such common instances are stored in "FOUTVECTOR". The process is described in Algorithm1.

Algorithm 1 : F_OUTVECTOR
Input: Models M_i and instances 'X'.
Output:FOUTVECTOR(Vector containing outliers of the model)

1. For each model 'M_i' in M

2. If S(X) ϵ C_j ' M_i then
3. Append 'C_j' to'X'
4. else
5. Add 'X' to OUT_i.
6. End if.
7. End if.
8. FOUTVECTOR=Intersection(OUT_1,OUT_2,...OUT_i);

5.1 Detecting concept- flow

The FOUTVECTOR generated in algorithm1, contains three types of outliers. The three outliers are caused due to concept flow, concept development and noise. The outliers are separated based on occurrence. The methodology to handle concept flow is discussed in "Handling concept-flow" section. The main task is to separate the instance based on causes. The concept-flow for an instance OUT_k from FOUTVECTOR, can be handled by using set of words S_k of the instance and comparing with set of words of different clusters belonging to the model M_i. The intersection operation performed on set S_k and set S_j of different classes C_j, and if result is more than 50 %, [3] it is declared as outlier due to "concept-flow". The instance is stored in "CONFLOW" vector.

5.2 Handling concept-flow

The "CONFLOW" vector consist of all the instance of results of step 3.3.1. To handle concept-flow the cluster or class to which the instance "OUT_k" initially belongs and word set "S_j" is found. Let "S_k" be set of words of an instance "OUT_k" present in "CONFLOW" vector. The difference function is performed on two sets S_j and S_k. The result set will represents set of latest words and stored in FLOWWORD vector along with class information from which instance is floated. A matrix CHKMAT[m,j] is constructed for class C_j of model M_i and distinct words, with all entries set to 0. The matrix is scanned and for each new flow word W_m of class C_j. For each incidence of new flow word W_m of class C_j, a value at location CHKMAT[m, j] is incremented by 1. For each unique word found a common threshold value is set for comparison. If the value at CHKMAT[m, j] is greater than the specified threshold value, then word Wm is appended to word set Sj of class C_j. This handles the concept flow and appropriately shift the new words in the existing classes. The process is explained in Algorithm2 below.

Algorithm 2 : CONCEPT_FLOW
Input: FVECTOR and Model 'M_i'
Output: CONFLOW(Instances having concept-flow and updated model)

1. For each OUT_k in FOUTVECTOR
2. for each cluster C_j in model M_i
3. result ←set-intersection(S_k, S_i)
4. if(¬(out (C_j)) and (s(result)>= (50% of C_j)))then
5. CONFLOW ← OUT_k
6. store information about C_j in jcount
7. end if
8. end for
9. end for
10. for each instance 'X' in CONFLOW belonging to cluster C_j
11. FLOWWORD ←set-difference(Xi,Sj)
12. end for
13. unique_flowword← Unique(FLOWWORD)
14. for each 'Wm' in Unique_flowword
15. for each class C_j in M_i
16. CHKMAT[m,j] ← CHKMAT[m,j]+1
17. end for
18. end for
19. if(CHKMAT[m,j]>Threshold)then
20. Append word Wm to'Sj' of class 'C_j'
21. end if
22. end algorithm

6. CONCEPT-DEVELOPMENT

6.1 Detecting concept-development

The concept development is detected by considering FOUTVECTOR. If an instance OUT_k, not satisfying the concept flow criteria, it is stated as concept development and stored in vector CONEVO. The instance is categorized as concept development, if more than 50 % of words of the instance does not suit the concept flow condition, of algorithm2. The threshold value 50 % is fixed based on the experiments carried out. In the conducted experiments it is examined that 50 % threshold is suitable for declaring instance as concept development.

6.2 Handling concept-development

The concept development is handled by creating a new class based on outcomes. The process of handling concept development also handles establishment of novel class. To produce new class or novel class, clustering algorithm is applied on CONEVO vector. The number of clusters is equal to number of classes in CONEVO vector. The clusters are appended to model M_i of the group.
Algorithm 3: CONCEPT_EVOLUTION
Input: FVECTOR and Model 'M_i'

Output: CONCEPT_EVOLUTION (vector having instances due to concept_evolution)

1. For each OUT_k in FOUTVECTOR
2. for each cluster C_j in model M_i
3. result ←set-intersection(S_k, S_i)
4. if((out (C_j)) and (s(result)< (50% of C_j)))then
5. CONCEPT_EVOLUTION ← OUT_k
6. end if
7. end for
8. end for
9.Apply K-medoid clustering on CONCEPT_ EVOLUTION
10. Obtain new clusters.
11. Append these new clusters to any of the previous models M_i.
12. end algorithm

7. DATA SETS

The algorithm proposed is only capable of handling the data which is not multi-labeled. Each occurrence present in the model only belongs to one class.

7.1 University data set

Initial experiment work was carried out on 4-University data set. After performing preprocessing, multi labeled and multi

valued data attributes were found in the data set. Since the designed algorithms were not able to handling multi labeled and multi valued features, the data set was not used for results production.

7.2 NASA aviation safety reporting system

NASA ASRS dataset holds the information about the different accidents that took place in the air industry. This data set is available online on NASA's official website. Each instance represents an accident and the possible reasons and outcomes. Each event has a related abnormality, and is considered as a different class, like Aircraft problem: not as much of severe, Aircraft problem: more serious, etc. The data also contains various multi-labeled and multi-valued characteristics. It also contained rows and columns having unfinished information.

The preprocessing step erased all such rows and columns form the data set. The data set contains six normal classes and two novel classes.

8. RESULT AND DISCUSSION

8.1 Techniques

Following techniques are used for relative study and results.

SCND: *It is an approach designed in the presented* research effort

O-F Approach: OLINNDA-FAE approach is the combination of OLINNDA Approach discussed in [10] and FAE approach discussed in [11]. In this combined approach OLINNDA works as a novel class identifier and FAE is used for classification. Mine-class is an approach developed by M. Masud et al. and is discussed in detail in [1]. MCM i.e. Multi Class Miner proposal is an approach developed by M. Masud et al. in [2].

8.2 Experiments

The experiments are based on subsequent assumption and system parameters.

i. Number of models in the group = 3

ii. Number of instances in chunk = 2000

In the data set choose for the experiments, no instance belonging to novel class was announced as existing class instance in the data set and very a small number of instances belonging to existing class were declared as novel class instance. The dataset also contains noise in the form of instances, belonging to existing class, but remained unclassified.

Table 1 shows the ERROR rate of model. The ERROR rate is defined as the percentage of not classified outliers in the data. Mis-classification represents percentage of instance not classified. False alarm rate represents percentage of instances wrongly classified.

Table 1. Summary of results

Data sheets	Error rate	Misclassification	False alarm rate (%age)
Datasheet 1	14.65	1.2	0.4
Datasheet 2	2.55	0.6	1
Datasheet 3	2.1	0.8	-

Table 2 running time (in seconds)

Approach	Running time (in seconds)
O-F Approach	141
Mine class	31.0
MCM	19.7
SCND	33.75(21+13)

Table 3 Comparison of results

Approach	Error(%age)	F_{new}	M_{new}
O-F	8.3	1.3	20.6
Mine class	17	1.1	8.4
MCM	1.8	0.68	0.7
SCND	6.4	1.1	-

The Table 2, explains the timing requirement of the system. The detail comparison with other approaches is presented in the Table 2. The time calculated is in seconds and is for one thousand instances of the dataset. In presented approach the most important portion of the time is utilized in loading the instances or data set i.e. about 20 s per thousand instances while the running time of the algorithm is about 13 s only. This is an indicator of decline in time required for classification. The relative study of experimental results gained with the previously developed approaches is presented in Table 3. Here ERROR is the total inaccuracy rate of the classifier. F_{new} is percentage of existing class instances defined as novel class instances. M_{new} is percentage of the novel class instances declared as existing class instances. In our approach

no novel class instance is declared as an existing class instance. In short, it can be concluded that other existing approaches had certain novel class instances that were classified as existing class instances but in research work presented in the paper the approach did not classify any novel class instance as an existing class instance. In Table 3 that M_{new} entry is empty. Also, the running time our developed algorithm is less than the running time of other techniques as shown in Table 2.

9. CONCLUSION AND FUTURE SCOPE

In this paper we try to propose a strategy based on string or pattern matching to handle data streams. The presented strategy can handle unlimited-length, concept-development and concept-flow. It can also detect multiple novel classes occurring simultaneously [12]. The presented strategy is based on string matching parameter instead of distance to handle the four challenges of data streams. The false alarm rate in the developed algorithm is quite little and can be believed as negligible. The presented strategy does not classify the novel class instance as existing class, but is not able to handle feature development effectively. the future scope of research work is to handle feature development effectively. All the experiments were carried out on equal size chunks, a future scope of the research is to verify the results on dynamic size chunks.

References

[1] Masud MM, Gao J, Khan L, Han J, Thuraisingham BM. Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Trans Knowl Data Eng. 2011;23(6):859–74.

[2] Masud MM, Gao J, Khan L, Han J, Thuraisingham BM. Classification and novel class detection in feature based stream data. IEEE Trans Knowl Data Eng. 2013;25(7):1484–97.

[3] Masud MM, Gao J, Khan L, Han J, Thuraisingham BM. "Integrating Novel class detection with classification for concept-drifting data streams," IEEE Trans Knowl Data Eng. 2009;25:7.

[4] Aggarwal CC, Han J, Wang J, Yu PS. A framework for on-demand classification of evolving data streams. IEEE Trans Knowl Data Eng. 2006;18(5):577–89.

[5] Masud MM, Gao J, Khan L, Han J, Thuraisingham BM. "Classification and novel class detection in data streams with active mining".

[6] Yang Y, Wu X, Zhu X. Combining proactive and reactive predictions for data streams. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. 2005. p. 710–15.

[7] Spinosa EJ, de Leon AP, de Carvalho F, Gama J. Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In: Proceedings of the 2008 ACM Symposium on Applied computing. 2008. p. 976–80.

[8] Masud MM, Chen Q, Gao J, Khan L, Han J, Thuraisingham BM. Classification and novel class detection of data streams in a dynamic feature space. In: Balcázar JL, Bonchi F, Gionis A, Sebag M, editors. Machine Learning and Knowledge Discovery in Databases. vol 6322. Berlin: Springer; 2010. p. 337–52.

[9] Bopche A, Nagle M, Gupta H. A review of method of stream data classification through optimized feature evolution process. Int J Eng Comput Sci. 2014;3(1):3778–83.

[10] OLINDDA: A cluster based approach for detecting novelty and concept-drift in data stream by Eduardo Spinosa J, Andr´e Ponce de Leon F, de Carvalho, Jo ao Gama in ACM Symposium of Applied Computing SAC'07.

[11] Wenerstrom B, Giraud-Carrier C. Temporal data mining in dynamic feature spaces. In: Data Mining, 2006. ICDM '06. Sixth International Conference on. Hong Kong:IEEE. 2006. p. 1141–45.

[12] Masud MM, Chen Q, Khan L, Aggarwal C, Gao J, Han J, Thuraisingham BM. Addressing concept-evolution in concept-drifting data streams. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. Sydney: IEEE; 2010. p. 929–34.

Author Profile

S.Mahammed Gouse[1] received the B.Tech. and M.Tech. degrees in Computer Science and Engineering from Jawaharlal Nehru Technological University, Anantapuram in 2009 and 2012, respectively. During 2012-2015 He has worked as a Assistant Professor in Balaji Institute of Engineering and Technology,Proddatur From 2015 – Till date working as a Assistant Professor in Avanthi Institute of Engineering and Technology, Hyderabad. He is a member of ISCA.

E.Amarnath Goud[2] received the B.Tech. And M.Tech. degrees in Computer Science and Engineering from Jawaharlal Nehru Technological University Hyderabad in 2012 and 2015, Working as a Assistant Professor in Avanthi's Scientific Technological & Research Academy ,Hyderabad. Interested Subjects are Computer Organization, Computer Networks. Doing Research on Big-Data.