Chandak *J Big Data* (2016) 3:5
DOI 10.1186/s40537-016-0040-9

Journal of Big Data
a SpringerOpen Journal

METHODOLOGY Open Access

Role of big-data in classification and novel class detection in data streams

M. B. Chandak*

*Correspondence:
hodcs@rknec.edu;
chandakmb@gmail.com
Department of Computer Science & Engineering, Ramdeobaba College of Engineering and Management, Nagpur, India

Abstract

"Data streams" is defined as class of data generated over "text, audio and video" channel in continuous form. The streams are of infinite length and may comprise of structured or unstructured data. With these features, it is difficult to store and process data streams with simple and static strategies. The processing of data stream poses four main challenges to researchers. These are infinite length, concept-evolution, concept-drift and feature evolution. Infinite-length is because the amount of data has no bounds. Concept-drift is due to slow changes in the concept of stream. Concept-evolution occurs due to presence of unknown classes in data. Feature-evolution is due to progression new features and regression of old features. To perform any analytics data streams, the conversion to knowledgable form is essential. The researcher in past have proposed various strategies, most of the research is focussed on problem of infinite-length and concept-drift. The research work presented in the paper describes a efficient string based methodology to process "data streams" and control the challenges of infinite-length, concept-evolution and concept-drift.

Subject areas Data mining, Machine learning

Keywords: Data stream, Data mining, Concept-drift, Concept-evolution, Novel, Features

Background

With the advancement of technology and use of internet of things [IoT], the amount of data generated over device communication channels is exponentially increasing. Data mining is one of the stream of "Database technologies" deals in processing large volume of structured and unstructured data. Initially, it was difficult to store and process the data generated over the communication channel, but in the present scenario researchers have developed methodologies to overcome the restriction. The data generated in text, audio, video format and is flowing from one network node to another, in un-interrupted fashion is denoted as "Data stream". The main characteristics of streaming data are: continuity, dynamic nature and no defined format. Its features keep on changing regularly, which makes it difficult to process. The four main challenges in processing streaming data are: infinite-length, concept-evolution, concept-drift and feature-evolution.

(i) The data stream is generated at very high speed and is infinite in size. It is impractical to store and process stream for training the systems. (ii) Concept drift is said to be

present when the underlying concept of the streams changes with time domain. This change causes the original classes of data, to drift towards the new features. (iii) Concept-evolution occurs when new classes evolve in the data. For example, concept evolution occurs when a new class of virus signature is detected or a new class of network attack is detected. Such evolution at run time are difficult to manage by any system. (iv) Feature evolution is the lengthy process. The new feature start appearing in the stream due to concept drift and concept evolution. The evolution of new feature fades away the existing features and over the period of time, substantial changes are observed in the system.

The static data classification techniques cannot be used in processing data streams due to four challenges [1–3]. There is a need to propose efficient classification techniques, which are suitable to handle the data stream challenges. The major challenges and solutions existing along with proposed solution is explained below:

a. Infinite length problem: Incremental learning models have been proposed by researchers in which hybrid batch incremental processing technique is used. The method splits the data streams into chunks of equal sizes, and training is provided to classification model to process the chunks.
b. Concept drift problem: This problem can be identified by monitoring the changes occurring in the streaming data. The changes occurring in the streaming data are variable and is handled by the data models which needs regular updates as per the changes in the streaming data. Most of researcher have provided partial solution, by fixing the classes of data. In the subsequent stage of research illustrated in the paper, the variability is modeled by varying the number of classes of data, which enables the system to process new class of data.
c. Concept evolution problem: In the presented research work, the concept evolution problem is addressed by allowing the classifier to mechanically detect the novel class, without having any prior training about the novel class. In past, traditional classifiers, were not able to detect novel classes, unless trained.
d. Feature evolution problem: This problem is address in the present research work by constantly monitoring of new features in the streaming data. The proposed solution is based on string comparison operation. The model design is collaboration 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-evolution, concept drift or noise. The new feature is used to update the existing model, to enable it to handle the challenges of streaming data.

Related work

In most of the research work in past, researcher have effectively solved the infinite length and concept drift problem, but no major findings are reported for concept evolution and feature evolution problems. For solving infinite length and concept drift 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 classification, 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 generated 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 observation of an abnormal distance measured between a data value and all other values of data in a random data sample. The problem of concept evolution and feature evolution can be solved by finding out outliers from the data samples. The outlier occurs in streaming data due to reasons like: noise, concept evolution or concept drift. The presented research work aims at finding the causes of occurrences of outliers. This will avoid misclassification of concept drift 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 evolution and concept drift, along with infinite length problem. In [7], methodology described uses clustering technique to detect novel classes. The clustering is performed on normal data contained in the region specified by hyper sphere. The model is continuously updated as the stream progress. When the cluster is formed and if it lies outside the hyper sphere, its density is checked 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 categorized as one class classifier and it assumes that shape of instances of normal class in feature space is convex, although it is not so in practical.

In the subsequent 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 compute distribution parameters. If an instance is not in-line with distribution parameter, it is classified as novel class. The non-parametric techniques are not based on data distribution and hence 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 detect presence of only one novel class. The research work presented in the paper describes an approach to detect multiple novel classes and is categorized as multiclass classifier.

Proposed approach

In the presented research work, the classifier operates on ensemble 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 an ensemble of models $\{M_1, M_2, M_3, \ldots M_n\}$. Following definitions are used in the proposed approach.

Definition 1 *Existing class* If a model Mi that belongs to an ensemble 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 ensemble M must be trained on class C.

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

Definition 3 *Outliers* If x is a test instance and if doesn't match the specifications of any of the class 'C' of the model Mi then 'x' is an outlier of the model Mi. The outliers don't belong to any of the class defined by the model.

Training phase

In the training phase the training data is divide 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 computed by applying K-medoid clustering technique on each chunk. The K-medoid technique performance is more suited to data set containing outliers [9]. The training phase will generate 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 (Si) defining the cluster. The classification rule followed by ensemble is: If 'X' is an instance to be tested, it is submitted to each model Mi in the ensemble to check whether it is an outlier for model Mi. If it is not an outlier (OUT), it will be classified by model Mi into one of its classes and if it is detected as an outlier by all the three models then it will be considered as a final outlier i.e. (FOUT).

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 Mi 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 'Mi'. If the test instance does not belong to any of the class defined by the model 'Mi', it is declared as an outlier (OUT) for that model 'Mi'.

Step 2 This step will find the final outliers of the ensemble. The outlier detected in step 1 for each model Mi 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.

Algorithm1. F_OUTVECTOR
Input: Models Mi and instances 'X'.
Output: FOUTVECTOR (Vector containing outliers of the model).

```
1: For each model 'Mi' in M
2:      If S(X) ∈ Cj'Mi then
3:          Append 'Cj' to 'X'.
4:      else
5:          Add 'X' to OUTi.
6:      End if.
7: End for.
8: FOUTVECTOR = Intersection (OUT1, OUT2…., OUTi);
```

Handling concept-drift

Detecting concept-drift

The FOUTVECTOR generated in algorithm1, contains three types of outliers. The three outliers are caused due to concept drift, concept evolution and noise. The outliers are separated based on occurrence. The methodology to handle concept drift is discussed

in "Handling concept-drift" section. The main task is to separate the instance based on causes.

The concept-drift 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-drift". The instance is stored in "CONDRIFT" vector.

Handling concept-drift

The "CONDRIFT" vector consist of all the instance of results of step 3.3.1. To handle concept-drift the cluster or class to which the instance "OUT_k" originally belongs and word set "S_j" is found. Let "S_k" be set of words of an instance "OUT_k" present in "CONDRIFT" vector. The difference operation is performed on two sets S_j and S_k. The result set will represents set of new words and stored in DRIFTWORD vector along with class information from which instance is drifted.

A matrix CHKMAT[mxj] is constructed for class Cj of model Mi and unique words, with all entries set to 0. The matrix is scanned and for each new drift word W_m of class C_j. For each occurrence of new drift word W_m of class C_j, a value at position 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 Cj. This handles the concept drift and properly shift the new words in the available classes. The process is explained in Algorithm2 below.

Algorithm2. CONCEPT_DRIFT
Input: FVECTOR and Model 'Mi'
Output: CONDRIFT (Instances having concept-drift and updated Model)

```
1: For each OUTk in FOUTVECTOR
2:        For each cluster Cj in model Mi
3:              Result ← Set-Intersection (Sk, Sj)
4:              If (⌐ (OUT (Cj)) and (S (Result) >= (50% of Cj))) then
5:              CONDRIFT ← OUTk.
6:             Store information about Cj in JCOUNT.
7:              End if
8:       End for.
9: End for.
10: For each instance 'X' in CONDRIFT belonging to cluster Cj
11:       DRIFTWORD ← Set-Difference (Xi, Sj)
12: End for.
13: Unique_driftword ← Unique (DRIFTWORD).
14: For each 'Wm' in Unique_driftword
15:       For each Class Cj in Mi
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 'Cj'
21: End if.
21: End algorithm.
```

Concept-evolution

Detecting concept-evolution

The concept evolution is detected by considering FOUTVECTOR. If an instance OUT_k, not satisfying the concept drift criteria, it is declared as concept evolution and stored in vector CONEVO. The instance is categorized as concept evolution, if more than 50 % of words of the instance does not satisfy the concept drift condition, of algorithm2. The threshold value 50 % is fixed based on the experiments conducted. In the conducted experiments it is observed that 50 % threshold is suitable for declaring instance as concept evolution.

Handling concept-evolution

The concept evolution is handled by creating a new class based on results. The process of handling concept evolution also handles creation of novel class. To generate 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 Mi of the ensemble.

Algorithm3: CON_EVOLUTION.
Input: FVECTOR and Model 'Mi'
Output: CONEVO (vector having instances due to concept_evolution)

1: For each OUT_k in FOUTVECTOR.
2: For each cluster C_j in model Mi
3: Result ← Set Intersection (S_k, S_j)
4: If (($OUT(C_j)$) and ($S(Result) < (50\% \text{ of } C_j)$)) then
5: CONEVO ← OUT_k.
6: End if
7: End for
8: End for
9: Apply K-medoid clustering on CONEVO.
10: Obtain new clusters.
11: Append these new clusters to any of the previous models Mi.
12: End algorithm

Data sets

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

University data set

Initial experiment work was carried out on 4-University data set. After performing preprocessing, multi labeled and multi valued data characteristics were found in the data set. Since the designed algorithms were not capable of handling multi labeled and multi valued features, the data set was not used for results generation.

NASA aviation safety reporting system

NASA ASRS dataset contains the information about the various 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 anomaly, and is considered as a different class, like Aircraft problem: less severe, Aircraft problem: more critical, etc. The data also contains various multi-labeled and multi-valued attributes. It also contained rows and columns having incomplete information. The preprocessing step deleted all such rows and columns form the data set. The data set contains six normal classes and two novel classes.

Result and discussion

Techniques

Following techniques are used for comparative study and results.

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

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 detector 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 scheme is an approach developed by M. Masud et al. in [2].

Experiments

The experiments are based on following assumption and system parameters.

i. Number of models in the ensemble = 3
ii. Number of instances in chunk = 2000

In the data set selected for the experiments, no instance belonging to novel class was declared as existing class instance in the data set and very few 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.

The Table 2, describes 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

Table 1 Summary of results

Datasheets	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 secs)
O-F Approach	141
Mineclass	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
MineClass	17	1.1	8.4
MCM	1.8	0.68	0.7
SCND	6.4	1.1	–

is for one thousand instances of the dataset. In presented approach the major 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 reduction in time required for classification.

The comparative study of experimental results obtained with the previously developed approaches is presented in Table 3. Here ERROR is the total error 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.

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 infinite-length, concept-evolution and concept-drift. 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 low and can be considered as negligible. The presented strategy does not classify the novel class instance as existing class, but is not able to handle feature evolution effectively. The future scope of research work is to handle feature evolution effectively. All the experiments were carried out on fixed size chunks, a future scope of the research is to check the results on dynamic size chunks.

Competing interests
The author declare that they have no competing interests.

Received: 16 November 2015 Accepted: 18 February 2016
Published online: 05 March 2016

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.