SIMILARITY AND ASSOCIATION KNOWLEDGE BASED RETRIVAL FOR CBR

S.Elango[1], P. Ganesh Kumar[2]

[1]PG Student, [2]Assistant Professor, Dept of TI , Anna University Coimbatore (India)

ABSTRACT

Data stream classification has been a widely studied research problem in recent years.The dynamic and evolving nature of data streams requires efficient and effective techniques that are significantly different from static data classification techniques. Two of the most challenging and well studied characteristics of data streams are its infinite length and concept-drift. Data stream classification poses many challenges to the data mining community. In this report, we address four such major challenges, namely, infinite length, concept-drift, concept-evolution, and feature-evolution.Since a data stream is theoretically infinite in duration, it is impractical to store and utilize all the historical data for training.Concept-drift is a usual phenomenon in data flows, which occurs as a result of changes in the underlying concepts.Concept-evolution comes about as a aftermath of new classes evolving in the current. Feature-evolution is a frequently occurring process in many streams, such as text streams, in which new features (i.e., words or phrases) appear as the stream progresses.

I. INTRODUCTION

The single model classification techniques apply some form of incremental learning to address the infinite length problem, and strive to adapt themselves to the most recent concept to address the concept-drift problem. Ensemble classification techniques maintain a fixed-sized ensemble of models, and use ensemble voting to classify unlabeled instances. These techniques address the infinite length problem by applying a hybrid batch-incremental technique.

Here the data stream is divided into equal sized chunks and a classification model is trained from each chunk. This model replaces one of the existing models in the ensemble, keeping the ensemble size constant. The concept-drift problem is addressed by continuously updating the ensemble with newer models, and striving to keep the ensemble consistent with the current concept. DXMiner also applies an ensemble classification technique. First, a decision boundary is built during training. Second, test points falling outside the decision boundary are declared as filtered outliers, or F-outliers. Finally, the F-outliers are analyzed to see if there is enough cohesion among themselves (i.e., among the F-outliers) and separation from the training instances.

we propose an improved technique to reduce both false alarm rate and increase detection rate. Our framework also allows for methods to distinguish among two or more novel classes.

We claim three major contributions in novel class detection for data streams . First, we propose a flexible decision boundary for outlier detection by allowing a slack space outside the decision boundary.

This space is controlled by a threshold, and the threshold is adapted continuously to reduce the risk of false alarms and missed novel classes. Second, we use a probabilistic approach to detect novel class instances using

the discrete Gini Coefficient. With this approach, we are able to distinguish different causes for the appearance of the outliers, namely, noise, concept-drift, or concept-evolves.

II. RELATED WORK

The author Charu C. Aggarwal stated that in recent years, the proliferation of VOIP data has created a number of applications in which it is desirable to perform quick online classification and recognition of massive voice streams. Typically such applications are encountered in real time intelligence and surveillance. In many cases, the data streams can be in compressed format, and the rate of data processing can often run at the rate of Gigabits per second.

The authors Charu C. Aggarwal, Senior Member, IEEE, Jiawei Han, Senior Member, IEEE, Jianyong Wang, Member, IEEE, and Philip S. Yu, Fellow, IEEE , Current models of the classification problem do not effectively handle bursts of particular classes coming in at different times. In fact, the current model of the classification problem simply concentrates on methods for one-pass classification modeling of very large data sets. Their example for data stream classification views the data stream classification problem from the point of persuasion of a dynamic approach in which simultaneous training and test streams are applied for dynamic sorting of information sets.This model reflects real life situations effectively, since it is desirable to classify test streams in real time over an evolving training and test current.

The authors, Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby RicardGavaldà stated that advanced analysis of data streams is quickly becoming a key field of data mining research as the number of applications demanding such processing increases.Online mining when such data streams evolve over time, that is when concepts drift or change altogether, is becoming one of the core subjects.When taking on non-stationary concepts, ensembles of classes have various advantages over single classifier methods: they are easy to scale and parallelize, they can adjust to change quickly by pruning under-doing components of the supporting players, and they therefore usually also generate more accurate concept descriptions.

III. SYSTEM ARCHITECTURE DESIGN

Design is the process of applying various techniques and principles for the purpose of designing a device. A process on a system is sufficient detail to permit its physical realization. It is a process through which requirements are translated in representation of the software.

From a project management point of view, software design is conducted in two steps. First one preliminary design that is concerned with the transformations of requirements into data and second step is software architecture detail design focus on refinement to architectural representation that lead to detail data structure and algorithmic representation of the software.

After detailed discussion with the user, the objectives requirements of data involved were identified. After performing necessary details of the document, output format and the frequency of reports are finalized with the user.

After analyzing the system flow, the file structure and logic of the programs are arrived and then the modification is verified. Design is concerned with identifying software component that specifying relationship among components. It specifies the software structure and provides the model for the implementation phase.

Techniques in the second category address the feature-evolution problem on top of the infinite length and concept-drift problems. It proposes a feature selection technique for data streams having dynamic feature space . Their technique consists of an incremental feature ranking method and an incremental learning algorithm.

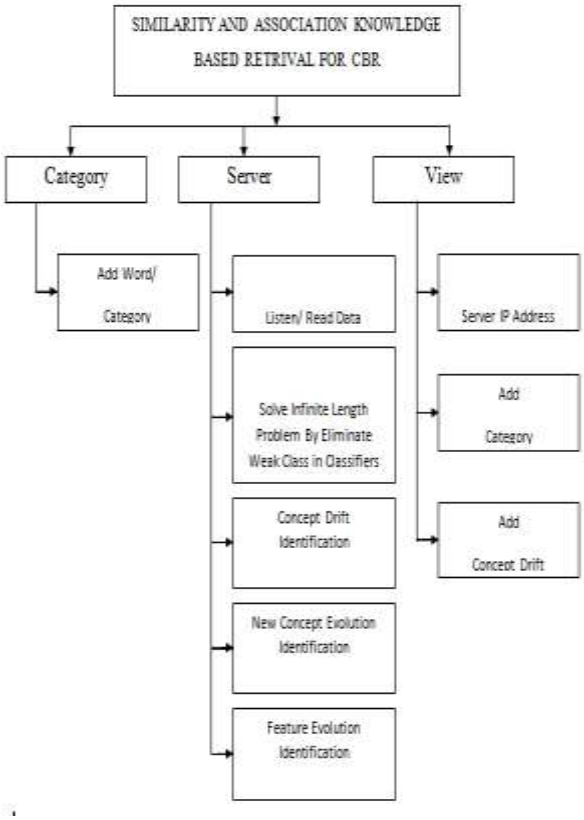


Fig 3.1Architecture Diagram

Techniques in the third category deal with the concept-evolution problem in addition to addressing the infinite length and concept-drift problems. An unsupervised novel concept detection technique for data streams is proposed, but it is not applicable to multi-class classification.

Our previous works Mine Class and DX Miner address the concept-evolution problem on a multiclass classification framework. They can detect the arrival of a novel class automatically, without being trained with any labeled instances of that class. However, they do not address the feature-evolution problem.

On the other hand, DX Miner addresses the more general case where features can evolve dynamically. Its effectiveness is shown analytically and demonstrated empirically on a number of real data streams.

IV. MODULE DESCRIPTION

The following modules are present in the project

- Solving Infinite Length Problem Module
- Concept Drift Identification Module
- Concept Evolution Identification Module
- Feature Evolution Identification Module

4.1 Solving Infinite Length Problem Module

When the data arrived is more and the classes formed out of them increases the problem is termed as infinite length problem. This is to be avoided. Each incoming instance in the data stream is first examined by an outlier

detection module to check whether it is an outlier. If it is not an outlier, then it is classified as an existing class using majority voting among the classifiers in the ensemble. If it is an outlier, it is temporarily stored in a buffer. When there are more new classes formed, then the classes with less content are discarded so that the number of classes is maintained within a given limit and this avoids the infinite problem.

4.2 Concept Drift Identification Module

The words and the category to which it belongs are added in the 'category' table. A client application is developed in which the text content is sent to the server application which updates the incoming message. The words are extracted and the words fell in the given category are identified and counted.

If there are more words in the category and the word count reduced in the successive incoming messages, then the concept is found to be reduced and when the number of words reduced to zero, the concept is said to be drifted. The number of observation time count is set so that when the number of word count is zero for that given number of time, then the concept is said to be drifted.

4.3 Concept Evolution Identification Module

During the concept evolution phase, the novel class detection module is invoked. If a novel class is found, the instances of the novel class are tagged accordingly. Otherwise, the instances in the buffer are considered as an existing class and classified normally using the ensemble of models. The words occurred frequently but not matched with any of the category available, and then the word is considered to be fallen in new class.

4.5 Feature Evolution Identification Module

In this module, along with concept evolution, feature evolution is identified. The repeated patterns are identified in the received messages and if it is found that more number of received messages contains the patterns, then it is said that feature evolution occurs.

V. INPUT DESIGN

- All the files from the disk should be acquired by data.
- It is suitable to more available data clearance and made available.
- The menu of design should be understandable and it is in the right format.

VI. DATABASE DESIGN

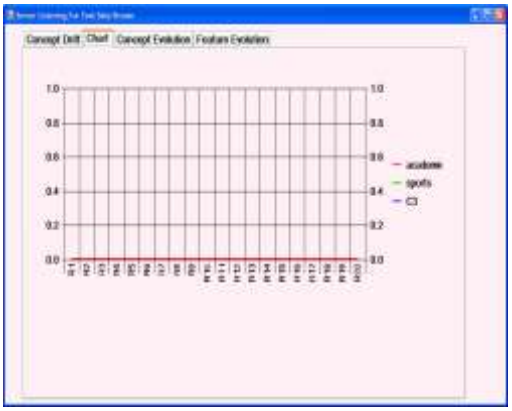
- Data Integration
- Data Integrity
- Data Independence

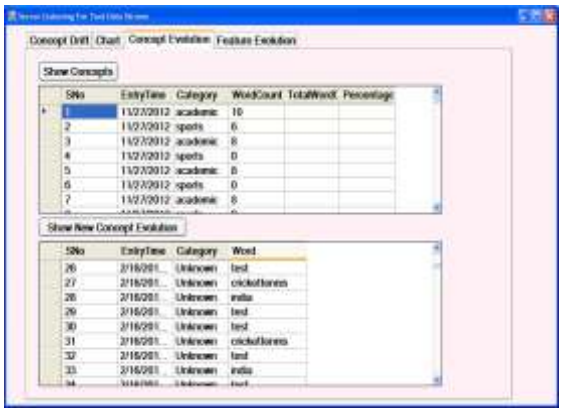
VII. OUTPUT DESIGN

- Output design generally refers to the results and information that are generated by the system for many end-users.
- The output is designed in such a way that it is attractive, convenient and informative.

VIII. RESULT

The project proposes a classification and novel class detection technique for concept-drifting data streams that addresses four major challenges, namely, infinite length, and concept-drift, concept-evolution, and feature evolution. The existing novel class detection techniques for data streams either do not address the feature-evolution problem or suffer from high false alarm rate and false detection rates in many scenarios





IX. CONCLUSION

Through this project, the drift detection issue is covered; Decision boundary for outlier detection is changing as the new data arrives; Uses any approach is used and so models with less importance are eliminated and space is provided for new models. The project considers the feature space conversion technique to address feature-evolution problem. Then, it identifies two key mechanisms of the novel class detection technique, namely, outlier detection, and identifying novel class instances

REFERENCES

[1] A. Aamodt and E. Plaza, “Case-based reasoning: Foundational issues, methodological variations, and system approaches,” AI Commun., vol. 7, pp. 39–59, Mar. 1994

[2] .R. Lopez De Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, . Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane,

[3] A. Aamodt, and I. Watson, “Retrieval, reuse, revision and retention in case-based reasoning,” Knowl. Eng. Rev., vol. 20, no. 3, pp. 215–240, 2005.

[4] Y. Guo, J. Hu, and Y. Peng, “Research on CBR system based on data mining,” Appl. Soft Comput., vol. 11, no. 8, pp. 5006–5014, 2011.

[5] Y.-J. Park, E. Choi, and S.-H. Park, "Two-step filtering datamining method integrating case-based reasoning and rule induction," Expert Syst. Appl., vol. 36, no. 1, pp. 861–871, 2009.

[6] B. Smyth and P. McClave, "Similarity vs. diversity," in Case-Based Reasoning Research and Development. Berlin, Germany: Springer- Verlag, 2001, pp. 347–361

[7] .J. L. Castro, M. Navarro, J. M. S´anchez, and J. M. Zurita, "Loss and gain functions for CBR retrieval," Inf. Sci., vol. 179, no. 11, pp. 1738–1750, 2009.

[8] H. Ahn and K.-J. Kim, "Global optimization of case-based reasoning for breast cytology diagnosis," Expert Syst. Appl., vol. 36, no. 1, pp. 724–734, 2009.

[9] B. Pandey and R. Mishra, "Case-based reasoning and data mining integrated method for the diagnosis of some neuromuscular disease," Int. J. Med. Eng. Informat., vol. 3, no. 1, pp. 1–15, 2011.

[10] Y.-B. Kang, A. Zaslavsky, S. Krishnaswamy, and C. Bartolini, "A knowledge-rich similarity measure for improving IT incident resolution process," in Proc. ACM Symp. Appl. Comput., 2010, pp. 1781–1788.