International Journal of Innovative Research in Engineering Science and Technology

APRIL 2018 ISSN 2320 –981X

THRESHOLD CO-OCCURRENCE PATTERNS ACROSS MULTIPLE STREAMS USING MINING

[1]Kalaivani.P , [2]Lavanya.P and [3]Malavika.T

[1]Department of Computer Science,
Vivekanandha College of Engineering for
Women, Tiruchengode-637 205,India.
mailtokalai1@gmail.com

[2]Department of Computer Science,
Vivekanandha College of Engineering for
Women, Tiruchengode-637 205,India.
lavan.sakthii@gmail.com

[3]Department of Computer Science,
Vivekanandha College of Engineering for
Women, Tiruchengode-637 205,India.
shanthmals@gmail.com

Corresponding Author:
Mr.P.Ramesh B.E.,M.E.,(Ph.D)
Department of Computer Science,
Vivekanandha College of Engineering for
Women, Tiruchengode-637205,India.
pramesh.swami@gmail.com
Contact: +91-7708197090

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 paper, we address four such major challenges, namely, infinite length, concept-drift, concept-evolution, and feature- evolution. Since a data stream is theoretically infinite in length, it is impractical to store and use all the historical data for training. Concept-drift is a common phenomenon in data streams, which occurs as a result of changes in the underlying concepts. Concept- evolution occurs as a result of new classes evolving in the stream. 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. Most existing data stream classification techniques address only the first two challenges, and ignore the latter two. The project proposes an ensemble classification framework, where each classifier is equipped with a novel class detector, to address concept-evolution.

KEYWORDS:

Map reduce,

Gini coefficient,

Threshold.

INTRODUCTION:

The use of Big Data is becoming a crucial way for leading companies to outperform their peers. In most industries, established competitors and new entrants alike will leverage data-driven strategies to innovate, compete, and capture value. Indeed, we found early examples of such use of data in every sector we examined. In healthcare, data pioneers are analyzing the health outcomes of pharmaceuticals when they were widely prescribed, and discovering benefits and risks that were not evident during necessarily more limited clinical trials. Other early adopters of Big Data are using data from sensors embedded in products from children's toys to industrial goods to determine how these products are actually used in the real world. Such knowledge then informs the creation of new service offerings and the design of future products. Big Data will help to create new growth opportunities and entirely new categories of companies, such as those that aggregate and analyse industry data. Many of these will be companies that sit in the middle of large information flows where data about products and services, buyers and suppliers, consumer preferences and intent can be captured and analyzed.

MapReduce is useful in a wide range of applications, including distributed pattern-based searching, distributed sorting, web link-graph reversal, Singular Value Decomposition, web access log stats, inverted index construction, document clustering, machine learning, and statistical machine translation. Moreover, the MapReduce model has been adapted to several computing environments like multi-core and many-core systems, desktop grids, volunteer computing environments, dynamic cloud environments, and mobile environments.

At Google, MapReduce was used to completely regenerate Google's index of the World Wide Web. It replaced the old *ad hoc* programs that updated the index and ran the various analyses. Development at Google has since moved on to technologies such as Percolator, Flume and MillWheel that offer streaming operation and updates instead of batch processing, to allow integrating "live" search results without rebuilding the complete index.

MapReduce's stable inputs and outputs are usually stored in a distributed file system. The transient data is usually stored on local disk and fetched remotely by the reducers.

The existing system includes three major contributions in novel class detection for data streams. First, it proposes 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, it applies a probabilistic approach to detect novel class instances using the discrete Gini Coefficient. With this approach, it is able to distinguish different causes for the appearance of the outliers, namely, noise, concept-drift, or concept-evolution. It derives an analytical threshold for the Gini Coefficient that identifies the case where a novel class appears in the stream. Third, it applies a graph-based approach to detect the appearance of more than one novel classes simultaneously, and separate the instances of one novel class from the others.

The paper proposed an ensemble classification framework, where each classifier is equipped with a novel class detector, to address concept-drift and concept-evolution. To address feature-evolution, we propose a feature set homogenization technique. The novel class detection module is used by making it more adaptive to the evolving stream, and enabling it to detect more than one novel class at a time for further reference. Comparison with state-of-the-art data stream classification techniques establishes the effectiveness of the proposed approach. The project proposes an ensemble classification framework, where each classifier is equipped with a novel class detector, to address concept-evolution. The project also enhances the novel class detection module by making it more adaptive to the evolving stream, and enabling it to detect more than one novel class at a time. In addition, concept drift approach is also applied.

METHODOLOGY:

The proposed system implements all existing system approach in addition with concept drift approach implementation. The basic steps in classification and novel class detection are as follows. Each incoming instance in the data stream is first examined by a 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.

Outlier Detection

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.

Concept Drift Identification

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.

Novel Class Detection

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.

Feature Evolution Identification

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.

Concept Evolution Identification

In this form, 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. A notify icon is displayed when new concept is evaluated.

Feature Evolution Identification

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. A notify icon is displayed when new concept is evolves.

Frequent Pattern Mining Across Multiple Databases

In this module frequent patterns that match the user-specified condition are mined. Mining sequential patterns across multiple databases with different domains has been addressed database. Assume two sequential databases D 1 and D 2, where D 1 and D 2 respectively have Xi and Xj at a given time. If Xi, Yi appears in more sequences than a user specified threshold, it is a frequent pattern.

Frequent Pattern Mining Across Multiple Streams

The module is continuous mining co-occurrence patterns that appear in at least *ρ* streams, where *ρ* is a user-specified threshold. Their empirical studies show that mining co-occurrence patterns across multiple streams is practically useful. The Seg-tree summarizes the valid transactions, and when a new transaction *t* arrives, *t* is inserted into the Seg-tree while merging the prefix nodes.

CP-Graph

In this module start with the CP-Graph structure and to update the answer quickly, it is desirable that user can efficiently enumerate necessary closed co-occurrence patterns and compute their counts.

The CP-Graph satisfies these requirements, and consists of *V* and *E*, where *V* (*E*) denotes the set of vertices (edges) at the current time- cycle *cnow*. In a nutshell, each object *oi*, which appears in the valid transactions, is regarded as a vertex *vi*, and edges are created between vertices, to represent patterns on the window. We below introduce the details of vertices and edges, and describe edges first, for ease of presentation. **(Figure 1)**

CONCLUSION:

The project identifies two key mechanisms of the novel class detection technique, namely, outlier detection, and identifying novel class instances, as the prime cause of high error rates for previous approaches.

To solve this problem, the project proposes an improved technique for outlier detection by defining a slack space outside the decision boundary of each classification model, and adaptively changing this slack space based on the characteristic of the evolving data. It also proposes a better alternative approach for identifying novel class instances using discrete Gini Coefficient and graph-based approach for multiple-novel class detection.

Through this project, the drift detection issue is covered; Decision boundary for outlier detection is changing as the new data arrives; Uses any drift detection technique to make the chunk size dynamic; Concept drift approach is used and so models with less importance are eliminated and space is provided for new models.

REFERENCES:

[1] C. C. Aggarwal. On classification and segmentation of massive audio data streams. *Knowl. and Info. Sys.*, 20:137–156, July 2009.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for on-demand classification of evolving data streams. *IEEE Trans. Knowl. Data Eng*, 18(5):577–589, 2006.

[3] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavald. New ensemble methods for evolving data streams. In *Proc. SIGKDD*, pages 139–148, 2009.

[4] S. Chen, H. Wang, S. Zhou, and P. Yu. Stop chasing trends: Discovering high order models in evolving data. In *Proc. ICDE*, pages 923–932, 2008.

[5] W. Fan. Systematic data selection to mine concept-drifting data streams. In *Proc. SIGKDD*, pages 128–137, 2004.

[6] J. Gao, W. Fan, and J. Han. On appropriate assumptions to mine data streams. In *Proc. ICDM*, pages 143–152, 2007.

[7] S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari. Adapted one-versus-all decision trees for data stream classification. *IEEE Trans. Knowl. Data Eng*, 21(5):624–637, 2009.

[8] G. Hulten, L. Spencer, and P. Domingos. Mining timechanging data streams. In *Proc. SIGKDD*, pages 97–106, 2001.

[9] I. Katakis, G. Tsoumakas, and I. Vlahavas. Dynamic feature space and incremental feature selection for the classification of textual data streams. In *Proc. ECML PKDD*, pages 102–116, 2006.

[10] I. Katakis, G. Tsoumakas, and I. Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 22:371–391, 2010.

[11] J. Kolter and M. Maloof. Using additive expert ensembles to cope with concept drift. In *Proc. ICML*, pages 449–456, 2005.

[12] D. D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[13] M. M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham. Classification and novel class detection of data streams in a dynamic feature space. In *Proc. ECML PKDD*,volume II, pages 337–352, 2010

[14] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. M. Thuraisingham. Addressing concept-evolution in concept-drifting data streams. In *Proc. ICDM*, pages 929–934, 2010.

LIST OF GRAPHS:Figure 1:

Data sets Graph