National Conference on Recent Advances in Computer Science and IT (NCRACIT)
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 1 | ISSN : 2456-3307

A Novel Approach to Extract Best-K Happening Patterns across Streams

k.Nithya[1], T.Aayebagavathi[2], V.Mahalakshmi[2], K.Nithya[2], K.Vijayalakshmi[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Nandha College of Technology, Erode-52, Tamil Nadu, India
[2]UG Students, Department of Computer Science and Engineering, Nandha College of Technology, Erode-52, Tamil Nadu, India
Knithiya89@gmail.com[1], mahacse001@gmail.com[2]

ABSTRACT

Frequent pattern mining is a fundamental problem for many domains, thus has a number of applications. In the Big data and IoT era, objects in these applications are often generated in a streaming fashion. An index-based algorithm is proposed in this project that addresses the challenge and provides the exact answer. The CP-Graph approach, a hybrid index of graph and inverted file structures. The CP-Graph computes the count of a given pattern and updates the answer while pruning unnecessary patterns.

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.

Keywords : Frequent Pattern Mining, Data Mining, Best-K Happening Patterns

I. INTRODUCTION

Data mining is about finding new information in a lot of data. Data mining, the extraction of hidden predictive information from large databases, it is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve.

It helps organization to make full use of the data stored in their databases and when it comes to decision making, this is true in all fields, and is also true in all different types of organizations. Knowledge discovery in databases (KDD) is the process of discovering useful knowledge from a collection of data. This widely used data mining

technique is a process that includes data preparation and selection, data cleansing, incorporating prior knowledge on data sets and interpreting accurate solutions from the observed results.

Data mining is the process of applying these methods to data with the intention of uncovering hidden patterns. It has been used for many years by businesses, scientists and governments to sift through volumes of data such as airline passenger trip records, census data and supermarket scanner data to produce market research reports. A primary reason for using data mining is to assist in the analysis of collections of observations of behavior. Such data are vulnerable to co linearity because of unknown interrelations. An unavoidable fact of data mining is that the sub sets of data being analyzed may not be representative of the whole domain, and therefore may not contain examples of certain critical relationships and behaviors that exist across other parts of the domain.

II. LITERATURE SURVEY

All known techniques for speaker voice analysis require the use of an offline training phase in which the system is trained with known segments of speech. The state-of-the-art method for text-independent speaker recognition is known as Gaussian Mixture Modeling (GMM), and it requires an iterative Expectation Maximization Procedure for training, which cannot be implemented in real time.

In this paper, they discussed the details of such an online voice recognition system. For this purpose, we use our micro-clustering algorithms to design concise signatures of the target speakers. One of the surprising and insightful observations from our experiences with such a system is that while it was originally designed only for efficiency, we later discovered that it was also more accurate than the widely used Gaussian Mixture Model (GMM).

This was because of the conciseness of the micro-cluster model, which made it less prone to over training. This is evidence of the fact that it is often possible to get the best of both worlds and do better than complex models both from an efficiency and accuracy perspective.

The problem of speaker voice analysis and classification is useful in a number of applications such as real time monitoring, detection, and surveillance. In this paper, we are concentrating on the problem of text-independent speaker classification in which the actual textual content of the speech is not available for modeling purposes. A number of statistical and machine learning methods have been recently proposed for speaker classification. Some examples of such techniques may be found in [1].

A well-known method for speaker classification and identification is that of Gaussian Mixture Modelling (GMM) [2]. The first step is to extract multi-dimensional feature vectors in order to represent portions of sampled speech. In this method, it is assumed that each data point extracted from the speech segments from a number of known speakers are used to estimate the parameters of a GMM model.

III. SYSTEM ANALYSIS – EXISTING SYSTEM

The existing system tackles the important problem of mining top-k closed co-occurrence patterns across multiple streams. To the best of our knowledge, this is the first work that addressees this problem. To support real-time update of the correct answer, the CP-Graph approach is proposed, an index that integrates graph and inverted file structures.

In addition to that, propose a CP-Graph traversal algorithm that exploits the apriori property to update the top-k closed co-occurrence patterns quickly and accurately. To prune unpromising patterns and update the answer incrementally, employed a strategy of summarizing the objects that appear in valid transactions by an in-memory index.

The existing system includes three major contributions in novel class detection for data

streams. A flexible decision boundary for outlier detection is applied 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. And 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. Finally it applies a graph-based approach to detect the appearance of more than one novel class simultaneously, and separate the instances of one novel class from the others.

IV. PROPOSED SYSTEM

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. When there are enough instances in the buffer, 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 ensemble of models is invoked both in the outlier detection and novel class detection modules. The outlier detection process utilizes the decision boundary of the ensemble of models to decide whether or not an instance is outlier. This decision boundary is built during training. The novel class detection process computes the cohesion among the outliers in the buffer and separation of the outliers from the existing classes to decide whether a novel class has arrived.

The proposed system enhances the existing novel class detection technique in three ways, which are:

- outlier detection using adaptive threshold,
- novel class detection and
- Simultaneous multiple novel class detection.

Advantages of Proposed System

The proposed system has following advantages.

- 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.

V. PROBLEM DEFINITION

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.

In recent years, advances in data storage technology have led to the ability to store the data for real-time transactions. Such processes lead to data which often grow without limit and are referred to as data streams. Discussions on recent advances in data stream mining may be found. One important data mining problem which has been studied in the context of data streams is that of classification. The

main thrust on data stream mining in the context of classification has been that of one-pass mining.

Online mining when such data streams evolve over time, that is when concepts drift or change completely, is becoming one of the core issues. When tackling non-stationary concepts, ensembles of classiers have several advantages over single classifier methods: they are easy to scale and parallelize, they can adapt to change quickly by pruning under-performing parts of the ensemble, and they therefore usually also generate more accurate concept descriptions.

Module Description

The following are the modules present in the project

- Frequent Pattern Mining Across Multiple Databases- (FPM-AM Database)
- Frequent pattern mining across multiple streams (FPMA-MS Stream)
- CP-Graph
- Solving infinite length problem
- Concept drift identification
- Concept evolution identification
- Feature evolution identification

VI.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.

VII. CONCEPT DRIFT IDENTIFICATION

In this form, 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.

VIII. SOURCE CODE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace ClassificationAndAdaptive
{
   public partial class frmMain : Form
   {
      public frmMain()
      {
         InitializeComponent();
      }
      private void
addWordCategoryToolStripMenuItem_Click(object
sender, EventArgs e)
      {
         frmWordCategory f = new
frmWordCategory();
         f.Show();
      }
      private void frmMain_Load(object sender,
EventArgs e)
      {
         this.Visible = false;
         frmLogin f = new frmLogin();
         f.ShowDialog();
         if (f.ok == true)
         {
```

```
            this.Visible = true;
        }
        else
        {
            Application.Exit();
        }
    }
    private void
listenToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        frmServerListen f = new frmServerListen();
        f.Show();
    }
    private void
addImageToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        frmAddImage f = new frmAddImage();
        f.Show();
    }
    private void
aboutTheProjectToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        MessageBox.Show(this.Text + ". Developed
using .Net Framework Version 2.0", "About",
MessageBoxButtons.OK,
MessageBoxIcon.Information);
    } } }
```

IX.CONCLUSION

The project proposes a classification and novel class detection technique for concept-drifting data streams that addresses four major challenges, namely, infinite length, 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.

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.

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.

X. 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.