Ensemble-Based Approach towards Concept Adapting Algorithms

Prof. Dipti D. Patil, Jyoti G. Mudkanna, Dr. Vijay M. Wadhai

.

Abstract: Most existing data mining classifiers cannot detect and classify the evolving class instances in real-time data stream mining problems. Unless and until the classification models are trained with the labeled instances of the evolving classes, classifiers are unable to discover new classes in real time. Different from data in traditional static databases, data streams typically arrive continuously in high speed with huge amount and changing data distribution. This raises new issues that need to be considered when developing classification techniques for data stream. This paper discusses those issues and challenges. The mining method of data streams needs to adapt to their changing data distribution; otherwise, it will cause the concept drifting problem. The paper describes features and analytical study of various ensemble-based concept adapting real time data stream mining algorithms those can play major role in making the real-time prediction.

Index Terms: algorithms, concept Drift, ensemble based classification, incremental Learning, analysis, comparative study, Stream mining,

---------- ◆ ----------

1 INTRODUCTION

Most existing data mining classifiers cannot detect and classify the evolving class instances in real-time data stream mining problems. For example, Weather conditions, economical changes, astronomical, and intrusion detection etc. Data mining plays a fundamental role in last few years. There are broad areas for the researchers. There exist emerging applications of data streams that require rule mining, classification. Different from data in traditional static databases, data streams typically arrive continuously in high speed with huge amount and changing data distribution. This raises new issues that need to be considered when developing classification techniques for data stream. The proposed work focuses on the classifiers. They should acknowledge the Non Stationary Environment (NSE). NSE may change at any time or may be continuously changing. The algorithm continually learns from the environment by constructing and organizing the knowledge base. If change has occurred, this change is learned. If change has not occurred, existing knowledge is re-enforced. The analytical study of different classifiers will be done after learning the algorithms.

1.1 AN INCREMENTAL LEARNING ALGORITHM

1) It should be able to learn additional information from new data.
2) It should not require access to the original data, used to train the existing classifier.
3) It should preserve previously acquired knowledge (It should not suffer from catastrophic forgetting).
4) It should be able to accommodate new classes that may be introduced with new data.

An algorithm that possesses these properties would be an indispensable tool for pattern recognition and machine learning researchers, since virtually unlimited number of applications can benefit from such a versatile incremental learning algorithm. The aim is to design a supervised incremental learning algorithm satisfying all of the above-mentioned criteria.[10]

The machine learning system thus needs to be able to detect when its knowledge is out of date and needs to be updated.

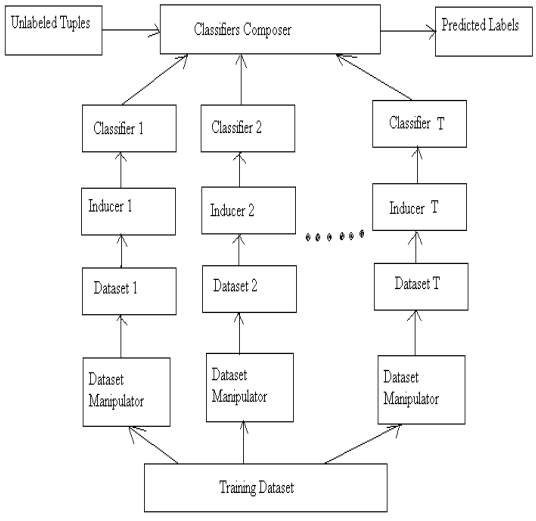

Figure1: Independent Ensemble Methodology [9]

1.1 CONCEPT DRIFT

Data stream mining has become a novel research topic of growing interest in knowledge discovery. Because of the high speed and huge size of data set in data streams, the traditional classification technologies are no longer applicable. In recent years a great deal of research has been done on this problem, most intends to efficiently solve the data streams mining problem with concept drift. The detection of changes in data streams is known to be a difficult task. When no information about the data distribution is available, an approach to cope with this problem is to monitor the performance of the algorithm by using the classification accuracy as a performance measure. The decaying of the predictive accuracy below a predefined threshold can be interpreted as a signal of concept drift. In such a case, however, the threshold must be tailored for the particular data set, since intrinsic accuracy can depends on background data. Furthermore, a naive test on accuracy does not take into account if the decrease is meaningful with respect to the past history. The proposed work is to track ensemble behavior by means of the concept of fractal dimension computed on the set of the most recent accuracy results.

2.1 WHAT IS CONCEPT DRIFT?

Streaming data environments are characterized by huge volumes of data flowing through a computer system. Typically, we don't have storage to retain all the data and we must learn its important characteristics and use them in future. Machine learning approaches assume a static underlying data distribution but this does not hold in streaming environments where data may span months and years and the generating sources may undergo periodic changes. For example, a customer's purchasing practices can change due to weather and economic cycles. In general, the generating sources may drift from one mode of operation to another. This type of change in a system's operating mode is known as concept drift. If there is a concept drift in the data and a fixed classification system continues to do classification then this system is bound to perform erroneously. So it is very important for the classification system to trace this concept drift and evolve to learn, and also use, the new concept.

2.2 HOW TO MANAGE CONCEPT DRIFT?

Basically, we need a way to identify in a timely manner those elements of the stream that are no longer consistent with the current concepts. A common approach is to use a sliding window. The intuition behind it is to incorporate new examples yet eliminate the effects of old ones. We can repeatedly apply a traditional classifier to the examples in the sliding window.

As new examples arrive, they are inserted into the beginning of the window; a corresponding number of examples are removed from the end of the window, and the classifier is reapplied. This technique, however, is sensitive to the window size, w. If w is too large, the model will not accurately represent the concept drift. On the other hand, if w is too small, then there will not be enough examples to construct an accurate model. Moreover, it will become very expensive to continually construct a new classifier model.

To adapt to concept-drifting data streams, the VFDT algorithm [5] was further developed into the Concept-adapting Very Fast Decision Tree algorithm (CVFDT). CVFDT [5] also uses a sliding window approach; however, it does not construct a new model from scratch each time. Rather, it updates statistics at the nodes by incrementing the counts associated with new examples and decrementing the counts associated with old ones. Therefore, if there is a concept drift, some nodes may no longer pass the Hoeffding bound. When this happens, an alternate sub tree will be grown, with the new best splitting attribute at the root. As new examples stream in, the alternate sub tree will continue to develop, without yet being used for classification. Once the alternate sub tree becomes more accurate than the existing one, the old sub tree is replaced.

Empirical studies show that CVFDT achieves better accuracy than VFDT with time-changing data streams. In addition, the size of the tree in CVFDT is much smaller than that in VFDT, because the latter accumulates many outdated examples.

2.3 A CLASSIFIER ENSEMBLE APPROACH TO STREAM DATA CLASSIFICATION

Let's look at another approach to classifying concept drifting data streams, where we instead use a classifier ensemble. The idea is to train an ensemble or group of classifiers (using, say, C4.5[7] or naïve Bays[9]) from sequential chunks of the data stream. That is, whenever a new chunk arrives, we build a new classifier from it. The individual classifiers are weighted based on their expected classification accuracy in a time-changing environment. Only the top-k classifiers are kept. The decisions are then based on the weighted votes of the classifiers.

3 CONCEPT DRIFT: NON-DETERMINISM

We can expect three possible scenarios when taking concept drift into consideration. The first is when the test set is not different in nature to the training set and we have no concept drift. The second is when there is some concept drift but with a majority of cases still being like the training cases. In this situation there will be a reduced number of cases classified but the accuracy should approximate the first scenario. The third scenario occurs when there is significant concept drift, here we would expect a large reduction in the number of cases classified with a concomitant fall in accuracy.

Finally, It's important to investigate a combined approach using both noise and concept drift approaches together.

3.1 LITERATURE SURVEY

for data stream mining and decision support is Many artificial intelligence researchers coming from different areas (data mining, machine learning, intelligent data analysis, pattern recognition, fuzzy logic, databases, etc.) are designing new approaches or writing new algorithms to adapt new classes in data streams. New challenge in mining is created by the rapid growth of information, the complexity and volume of data in particular. The main issue is that, this data production often takes the form of high-speed continuous flow of data.

Ensembles of learning machines have been used to learn in the presence of concept drift. There has been no deep study of why the ensembles are helpful for learning the concept drift? Which of their features can contribute to detect concept drift? What are the best methods to find out the drift? To deal with these arising questions and finding out the correct solution is the challenging task.[5] Dealing with continuous, and possibly infinite, flows of data require different approaches for data processing, machine learning and knowledge discovery. Particular issues regarding to the real time data stream include summarization of infinite data, incremental learning, resource-awareness, real-time monitoring of stream and to track the changes and recurrences in stream etc. This is an incremental task which requires incremental algorithms to integrate very large databases. Streaming artificial intelligence is progressively more important in the real time data stream research domain, as new algorithms are needed to process any type of data in reasonable time. Adoption and development of customized techniques still to come.

The goal of this topic is to present cutting-edge research in data stream processing in interested applications and timely analysis of data streams for any decision support.

Decision support, alerting services, ambient intelligence, assisted leaving and personalization services are just few examples of expected uses of actionable knowledge extracted from data streams. All of them are characterized by the high-speed at which huge amounts of data are produced, and often require fast and accurate information retrieval and analysis, that can effectively support critical and important decisions.[6]

4 LEARNING CONCEPT DRIFT WITH ONLINE ENSEMBLE APPROACH

Ensemble methods maintain a collection of learners and combine their decisions to make an overall decision.

. The online ensemble learning approaches used to learn concept drift are as follows:

I. Online Bagging .
II. DDD
III. Learn++.NSE

4.1 ONLINE BAGGING [3]

Algorithm 1 Online Bagging

Inputs: ensemble h; ensemble size M; training example d; and online learning algorithm for the ensemble members OnlineBaseLearningAlg;
1: for m ← 1 to M do
2: K ← Poisson(1)
3: while K >0 do
4: hm ← OnlineBaseLearningAlg(hm, d)
5: K ← K − 1
6: end while
7: end for
Output: updated ensemble h

4.2 DDD (DIVERSITY FOR DEALING WITH DRIFTS) [2]

Algorithm 2 DDD

Inputs:
• multiplier constant W for the weight of the old low diversity ensemble;
• online ensemble learning algorithm EnsembleLearning;
• parameters for ensemble learning with low diversity pl and high diversity ph;
• drift detection method DetectDrift;
• parameters for drift detection method pd;
• data stream D;
1: mode ← before drift
2: hnl ← new ensemble /* new low diversity */
3: hnh ← new ensemble /* new high diversity */
4: hol ← hoh ← null /* old low and high diversity */
5: accol ← accoh ← accnl ← accnh ← 0 /* accuracies */
6: stdol ← stdoh ← stdnl ← stdnh ← 0 /* standard deviations */
7: while true do
8: d ← next example from D
9: if mode == before drift then
10: prediction ← hnl(d)
11: else
12: sumacc ← accnl + accol *W + accoh
13: wnl = accnl/sumacc
14: wol = accol * W/sumacc
15: woh = accoh/sumacc
16: prediction ← WeightedMajority(hnl(d), hol(d), hoh(d), wnl, wol, woh)
17: Update(accnl, stdnl, hnl, d)
18: Update(accol, stdol, hol, d)
19: Update(accoh, stdoh, hoh, d)
20: end if

21: drift ← DetectDrift(hnl, d, pd)
22: if drift == true then
23: if mode == before drift OR
(mode == after drift AND accnl > accoh) then
24: hol ← hnl
25: else
26: hol ← hoh
27: end if
28: hoh ← hnh
29: hnl ← new ensemble
30: hnh ← new ensemble
31: accol ← accoh ← accnl ← accnh ← 0
32: stdol ← stdoh ← stdnl ← stdnh ← 0
33: mode ← after drift
34: end if
35: if mode == after drift then
36: if accnl > accoh AND accnl > accol then
37: mode ← before drift
38: else
39: if accoh − stdoh > accnl + stdnl AND accoh − stdoh > accol + stdol then
40: hnl ← hoh
41: accnl ← accoh
42: mode ← before drift
43: end if
44: end if
45: end if
46: EnsembleLearning(hnl, d, pl)
47: EnsembleLearning(hnh, d, ph)
48: if mode == after drift then
49: EnsembleLearning(hol, d, pl)
50: EnsembleLearning(hoh, d, pl)
51: end if
52: if mode == before drift then
53: Output hnl, prediction
54: else
55: Output hnl, hol, hoh, wnl, wol, woh, prediction
56: end if
57: end while

4.3 LEARN++.NSE [1]

Algorithm 3 Learn++.NSE
Input: For each dataset Dt t = 1,2, ...
Training data {xt (i) ∈ X; yt (i) ∈ Y = {1, ... , c}}, i = 1, ... , mt
Supervised learning algorithm BaseClassifier
Sigmoid parameters a (slope) and b (infliction point)
Do for t = 1,2, ...
If t = 1, Initialize D1 (i) = wt(i) = 1/m1, ∀i,
(1)
Go to step 3. Endif

1. Compute error of the existing ensemble on new data
Et = $\sum_{i=1}^{mt} 1/m^t$. [H^{t-1} (xt (i)) ≠ yt (i)_
(2)

2. Update and normalize instance weights
wt = . 1/ mt Et, Ht−1 (xt (i)) = yt (i)
1, otherwise
(3)

Set Dt = wt/_mt wt(i) ⇒Dt is a distribution
(4)
3. Call BaseClassifier with Dt, obtain ht: X →Y

4. Evaluate all existing classifiers on new data Dt
$\varepsilon_k^t = \sum_{i=1}^{m^t} D^t$ (i) [hk (xt (i)) ≠ yt (i)] for k = 1, ..., t
(5)

If $\varepsilon_{k=t}^t$ >1/2, generate a new ht.
If $\varepsilon_{k<t}^t$ >1/2, set ε_k^t = 1/2,
$\beta_k^t = \varepsilon_k^t/(1- \varepsilon_k^t)$, for k = 1, ..., t → 0 ≤β_k^t ≤ 1
β_k^{t-j} (6)

5. Compute the weighted average of all normalized errors for kth classifier hk: For a,b ∈R

$\omega t = 1/(1 + e^{-a(t-k-b)})$, $\omega_k^t = \omega_k^t / \sum_{j=0}^{t-k} \omega_k^{t-j}$
(7)

$\beta_k^t = \sum_{j=0}^{t-k} \omega_k^{t-j} \beta_k^{t-j}$, for k = 1, ..., t
(8)
6. Calculate classifier voting weights
W_k^t = log(1/ β_k^t), for k = 1, ..., t
(9)
7. Obtain the final hypothesis
Ht (xt (i)) = arg maxc $\sum_k W_k^t$. [hk (xt (i)) = c]
(10)
In the averaging ensemble for scalable learning over very-large datasets [12], a model's performance can be estimated before it is completely learned [10][11]. Weighted ensemble classifiers on concept-drifting data streams are used in the LEARN++.NSE . It combines multiple classifiers weighted by their expected prediction accuracy on the current test data. Compared with incremental models trained by data in the most recent window, this approach combines talents of set of experts based on their credibility and adjusts much nicely to the underlying concept drifts. Also, the dynamic classification technique [9] to the Concept-drifting streaming environment is introduced. It enables us to dynamically select a subset of classifiers in the ensemble for prediction without loss in accuracy.

4.4 ANALYSIS OF ALGORITHMS:

TABLE 1
COMPARATIVE STUDY OF ALGORITHMS

Sr. No.	Algorithm Name	Feature	Accuracy
1.	Online Bagging[3]	Determine	To obtain

		the role of diversity by itself in the presence of drifts	lower test errors
2.	DDD [2]	More robust to false alarms (false positive drift detections) and faster recovery from drifts	Best prequential accuracy
3.	LEARN++.NSE[1]	Can track the changing environments very closely, regardless of the type of concept drift.	Higher Accuracy for NSE

5 ADVANTAGES

1. Classifier ensembles offer a significant improvement in prediction accuracy [13][2].
2. Building a classifier ensemble is more efficient than building a single model, since most model construction algorithms have super-linear complexity.
3. The nature of classifier ensembles lend themselves to scalable parallelization [12] and on-line classification of large databases [4].
4. Stream mining is particularly significant for applications that need real-time analysis of continuous data streams.[13]

6 APPLICATIONS

1. Health Care Applications
2. Industrial process control
3. Computer security
4. Intelligent user interfaces
5. Market-basket analysis
6. Information filtering
7. Prediction of conditional branch outcomes in microprocessors

7 CONCLUSION AND FUTURE SCOPE

The problem of concept drift and different approaches to adapt it is discussed in this paper. To summarize, three basic approaches to handling concept drift can be distinguished: instance selection, instance weighting, and ensemble learning. Real data including different types of concept drift are needed to experiment with proposed approaches to validate them and check their robustness to the change of different data characteristics and, in particular, their scalability. An important part of the research on concept drift is developing criteria for detecting crucial changes that allow adapting the model only if inevitable. Currently suggested "triggers" are not robust to different types of concept drift and different levels of noise, and more research is needed in this direction. The work focused towards developing such an methodology to detect and adapt this concept drift gradually and develop an efficient online learner for mining real time data streams.

ACKNOWLEDGMENT

This is the research work associated to the PhD work going on Under Sant Gadge Baba Amravati University (SGBAU) and PG Department of Computer Science and Engineering, Amravati. My sincere thanks to the centre Head Dr. V. M. Thakre and the faculty for their extensive support.

REFERENCES

[1] Ryan Elwell, and Robi Polikar, "Incremental Learning of Concept Drift in Non- stationary Environments" IEEE Transactions On Neural Networks, Vol. 22, No. 10, Pp. 1517-1531, October 2011.

[2] Leandro L. Minku and Xin Yao, "DDD: A New Ensemble Approach For Dealing With Concept Drift" IEEE transactions on knowledge and data engineering, 2011.

[3] Leandro L. Minku, Allan P. White, and Xin Yao, "The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift" IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 5, pp. 730 – 742,May 2010.

[4] MAHNOOSH KHOLGHI "An Analytical Framework For Data Stream Mining Techniques Based On Challenges And Requirements" International Journal of Engineering Science and Technology IJEST , Mar 2011.

[5] Sasthakumar Ramamurthy, Raj Bhatnagar " Tracking Recurrent Concept Drift Streaming data using Ensemble Classifiers" Machine Learning and Applications, ICMLA 2007.

[6] Dengyuan Wu , Ying Liu ; Ge Gao ; Zhendong Mao ; Weishan Ma ; Tao He " An adaptive ensemble classifier for concept drifting stream" Computational Intelligence and Data Mining, Pp. 69 – 75, 2009.

[7] Mohamed Medhat Gaber, Shonali Krishnaswamy and Arkady Zaslavsky "Cost Efficient Mining Techniques for Data Streams " Australia Conferences in Research and Practice in Information

Technology, Vol. 32. 2004.

[8] Mahesh Kr. Singh, Zaved Akhtar, Devesh Kr Sharma "Challenges and Research Issues in Association Rule Mining" in the proc. of International Journal of Electronics and Computer Science Engineering (IJECSE) V1N2 pp. 767-774, 2006.

[9] Lior Rokach "Ensemble-based classifiers" Springer Science + Business Media B.V. 19 pp. 1-39 November 2009. .

[10] Robi Polikar, Lalita Udpa, Satish S. Udpa and Vasant Honavar "Learn++: An Incremental Learning Algorithm for supervised Neural Networks" IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 31, No. 4, pp. 497-508 November 2001.

[11] Michael Harries Kim Horn " Detecting Concept Drift in Financial Time Series Prediction using Symbolic Machine Learning "

[12] Gary R. Marrs, Ray J. Hickey, Michaela M. Black "Modeling the Example Life- Cycle in an Online" Classification Learner Proceedings of the First International Workshop on HaCDAIS held in conjunction with ECML/PKDD Pp. 57-64 September 24, 2010.

[13] Haixun Wang Wei Fan Philip S. Yu 1Jiawei Han" Mining Concept Drifting Data Streams using Ensemble Classifiers" 2002.

[14] Evguen Smirnov "Ensemble Classifiers" www.unimaas.nl / datamining / Slides2011.

Author's Information:

- *Dipti D.Patil is currently pursuing Ph. D. degree program in Computer Engineering, from Sant Gadgebaba Amravati University, India Email: dipti.dpatil@yahoo.com*
- *Jyoti G. Mudkanna is currently pursuing Masters degree in Computer Engineering from MAEER's MIT College of Engineering, Pune. India. Email: jmudkanna@gmail.com*
- *Vijay M. Wadhai is currently working as Principal, MAEER's MIT College of Engineering Pune, India. Email: wadhai.vijay@gmail.com*