Complex Class Classification for Gradually Novel Classes in Data Stream Mining

Mr. Prashant M. Gore
PG Student
Department of Computer Engineering
PVPIT Savitribai Phule Pune University, India

Prof. S. V. Bodake
Assistant Professor
Department of Computer Engineering
PVPIT Savitribai Phule Pune University, India

Abstract

Data Stream Mining is the process of extracting knowledge structures from continuous, rapid data records. Now a day's huge amount of data is processed & analyzed. So it is very important to classify data & information properly. The information is basically unstructured & continuous. So huge volume of continuous data which has multidimensional feature & often fast changing. It is required to construct model which adapt such changes & give fast response. Such information flow examples are network traffic, sensor data, call center records etc. Class evolution is now a day's important topic in data stream mining which handles such data. So in previous work proposed a model Class Based ensemble for Class evolution (CBCE) to maintain such a large amount of streams. But for complex & massive data result would be different. So complex class ensemble model (CCEM) is proposed for classification so huge & complex classes can be handled & classify & also proposed a model for class disappearance only so that more emphasize on class disappearance than class reoccurrence & novel class.

Keywords: Data Stream Mining, Class Evolution, Ensemble Model, Incremental Learning

I. INTRODUCTION

Generally data stream mining basically referred when infinite number of data elements is arrived rapidly. Such examples are telecommunication calling records, credit card transaction flow, network monitoring & traffic engineering, financial market, audio & video recording of various processes, computer network information flow etc. In such cases data is distributed which is dynamic always. Data distribution is always changing so called concept drift. In data streaming one pass data & processing data depends on rate of incoming data. Concept drift is a phenomenon describing change in data distribution, character or their meaning e.g. assigning emails to the "spam" category. In data stream the joint probability distribution can vary over time. Concept drift can be sudden, incremental, gradual, recurring context. For example the behavior of the customers in an online shop may change over time. For example, if weekly merchandise sales are to be predicted, and a predictive model has been developed that works satisfactorily. The model may use inputs such as the amount of money spent on advertising, promotions being run, and other metrics that may affect sales. The model is likely to become less and less accurate over time - this is concept drift. In the merchandise sales application, one reason for concept drift may be seasonality, which means that shopping behavior changes seasonally. Perhaps there will be higher sales in the winter holiday season than during the summer. So Class evolution having major impact on data stream mining. The model is developed which classify data in three models namely class emergence, class reoccurrence & class disappearance. In some literature class evolution is also called class incremental learning. Class emergence implies new class evolved in data streams e.g. Currency announcement by Prime Minister of India & its different comments on twitter so its class emergence or novel class. Class reoccurrence implies the classes which reoccur again depend on timestamp. Class disappearance implies existing class would not be received in next time stamp. In previous studies authors basically focus on novel class & class reoccurrence only. But Data mining task of large & massive complex class is still difficult & one of the research topic. So proposed system mainly emphasize on this data only which is large & complex. Also proposed model focuses on Class disappearance as well which is not occurred in previous studies.

Complex class Ensemble Model (CCEM) is proposed which first evaluate number of disappearance classes which depends on timestamp & remove it from complex set of classes. The model calculates arrival time of remaining classes & assigns unique ID to each class. CCEM model also calculates prior probabilities of each class. So every time this model updates after arrival of new examples.

II. LITERATURE SURVEY

Concept drift is now a day's research topic in data stream mining. Due to concept drift dynamic imbalance problem occurs. So different learning methodologies are proposed in paper[1][2].In this paper different kinds of leanings are proposed like batch mode learning, incremental learning, online learning & anytime learning. Also they proposed how to deal with concept drift.

A sliding window approach is proposed in paper[3]which stores in memory number of recent examples. The size of window can be static or dynamic. In this model the data in memory is updated every time & data affected by concept drift are removed. But sometimes it also forgets useful information. so this is disadvantage of this model so results are not reliable & prediction cannot

be determined. a new approach for dealing with distribution change and concept drift when learning from data sequences that may vary with time. use sliding windows whose size, instead of being fixed a priori, is recomputed online according to the rate of change observed from the data in the window itself: The window will grow automatically when the data is stationary, for greater accuracy, and will shrink automatically when change is taking place, to discard stale data. This delivers the user or programmer from having to guess a time-scale for change.W. Nick Street &YongSeog Kim [4] proposed streaming ensemble algorithm. Ensemble methods have recently gathered a great deal of attention in the machine learning community. Techniques such as Boosting & Bagging have proven to be highly effective but require repeated resampling of the training data making them inappropriate in a data mining context. The methods presented in this paper take advantage of plentiful data, building separate classifiers on sequential chunks of training points. These classifiers are combined into a fixed size ensemble using a heuristic replacement strategy. The result is a fast algorithm for large-scale or streaming data that classifies as well as a single decision tree built on all the data requires approximately constant memory & adjust quickly to concept drift.

Matthew Karnick, Metin Ahiskali proposed [5] concept drift in non-stationary environments using an ensemble of classifier approach. Specifically in this paper generate a new classifier using each additional dataset that becomes available from the changing environment. The classifiers are combined by modified weighted majority voting, where the weights are dynamically updated based on the classifiers' current and past performances, as well as their age. This mechanism allows the algorithm to track the changing environment by weighting the most recent and relevant classifiers higher. However, it also utilizes old classifiers by assigning them appropriate voting weights should a cyclical environment renders them relevant again. The algorithm learns incrementally, i.e., it does not need access to previously used data. The algorithmic also independent of a specific classifier model, and can be used with any classifier that fits the characteristics of the underlying problem. We describe the algorithm, and compare its performance using several classifier models, and on different environments as a function of time for several values of rate-of-change.

Mohammad M. Masud, Jing Gao[6] proposed integrating novel class detection with classification for concept drift data streams. In a typical data stream classification task, it is assumed that the total number of classes is fixed. This assumption may not be valid in a real streaming environment, where new classes may evolve. Traditional data stream classification techniques are not capable of recognizing novel class instances until the appearance of the novel class is manually identified, and labeled instances of that class are presented to the learning algorithm for training. The problem becomes more challenging in the presence of concept-drift, when the underlying data distribution changes over time. We propose a novel and efficient technique that can automatically detect the emergence of a novel class in the presence of concept-drift by quantifying cohesion among unlabeled test instances, and separation of the test instances from training instances. Our approach is non-parametric, meaning; it does not assume any underlying distributions of data. Comparison with the state-of-the-art stream classification techniques proves the superiority of our approach.

III. EXISTING SYSTEM

In today's information society computer users are used to gathering & sharing data anytime. This concerns application such as social networks, banking, and telecommunication, health care, research & entertainment etc. As a result a huge amount of data related to all human activity is gathered for storage & processing purposes. These data sets may contain interesting and useful knowledge represented by hidden patterns, but due to the volume of the gathered data it is impossible to manually extract that knowledge. That is why data mining and knowledge discovery methods have been proposed to automatically acquire interesting, non-trivial, previously unknown and ultimately understandable patterns from very large data sets Typical data mining tasks include association mining, classification, and clustering, which all have been perfected for over two decades. A data stream is an ordered sequence of instances that arrive at a rate that does not permit to permanently store them in memory. Data streams are potentially unbounded in size making them impossible to process by most data stream mining approaches.

The distribution generating the items of a data stream can change over time. These changes, depending on the research area, are referred to as temporal evolution, covariate shift, non-stationary, or concept drift. Concept drift is an unforeseen substitution of one data source with another source.

Concept drift also occurs in class evolution. So existing data streams are classified as different kinds of classes. Class emergence, class reoccurrence & class disappraence. Class emergence implies new classes which generates day by day .i.e. unknown class received at current time stamp. Class disappearance indicates the class which would not occur again. Class reoccurrence indicates that class can recurs again in next time stamp. Concepts drift mainly impact on prior probability which may change due to change in data distribution. So in existing system Class Based Ensemble for Class evolution (CBCE) model is proposed which process data streams chunk by chunk & build base learner for each chunk. CBCE maintains base learner for every class &updates base learner every time as shown in fig.

In CBCE when new data arrive it divided into chunks & predicts its labels using ensemble approach. After obtaining the true label model is updated. The model calculates prior probability of class. If prior probability is small it is considered as class disappearance. In this case it is conserved. If disappearance class reoccurs then it is reactivated again and calculates prior probability re-estimated from current data.

Since CBCE model is maintained for certain class & it is flexible i.e. can be removed or created. For massive-volume data streams the master-slave data structure is used.

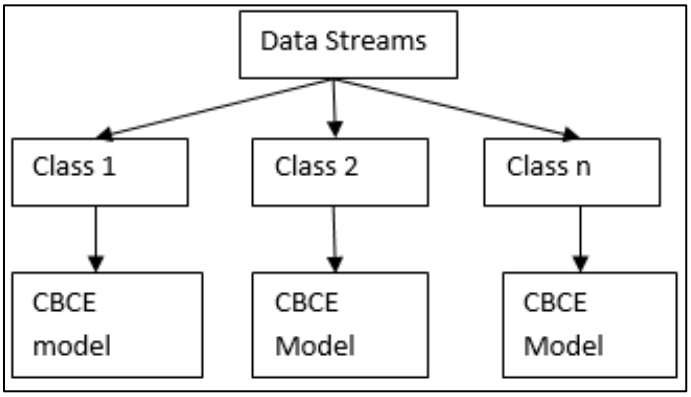

Fig. 1: CBCE Model

Class evolution is considered as gradual process i.e. sizes can be increased or shrink. CBCE can adapt well in all class i.e. Class emergence, class reoccurrence & class disappearance. CBCE handles dynamic imbalance problem. Class reoccurrence means that an example with the label of a disappeared class is received again. Effective handling of class reoccurrence could make use of past training efforts. For the inactivated CB model of a disappeared class, it can be used again for classification when an example with an old label arrives, which makes CBCE efficient. Once class reoccurrence happens, the model re-estimates the prior probability in the same way as class emergence, and activates the CB model in classification. In CBCE, the change of class-conditional probability distribution means that the CB model is no longer able to correctly identify its corresponding positive examples. To handle this problem, a simple and yet effective drift detection method, DDM is applied to check a CB model's validity. If a CB model was significantly affected by this type of change, it would be re-initialized. Each example used for training the CB model is incorporated for the detection of the change. If the warning level is reached, the CBmodel is likely to be outdated and the following sampled examples are stored. If DDM detects a drift in a CB model, the model is re-initialized by these examples. Through this method, the likelihood value obtained with each CB models avoided to be affected by the change of class-conditional probability distribution.

In CBCE, when a new example is received, the ensemble model first predicts its label for practical use. After obtaining the true label of this example, each CB model is updated to track the up-to-date concept. If a novel class emerges, a new CB model corresponding to this class is initialized. A sufficiently small prior probability of a class implies its disappearance. In this case, the corresponding CB model is inactivated but still conserved. If a disappeared class reoccurs, the corresponding CB model will be re-activated with the prior probability of the class being re-estimated from the current data. In order to handle the dynamic class imbalance problem caused by the gradual process of class evolution, CB models use under-sampling with a dynamic probability to sample the examples to balance the training data. It is noted that all active CB models are used for classification, with decision determined by choosing a class whose CB model outputs the highest score. A change detection method is used to monitor changes in the class conditional probability distributions corresponding to each CB model. If a change is detected, the corresponding CB model is reset.

Previous investigations on data stream mining assume class evolution to be the transient changes of classes, which does not hold for many real-world scenarios. In this work, class evolution is modeled as a gradual process, i.e., the sizes of classes increase or shrinks gradually. A new data stream mining approach, CBCE, is proposed to tackle the class evolution problem in this scenario. CBCE is developed based on the idea of a class-based ensemble. Specifically CBCE maintains a base learner for each class and updates the base learners whenever a new example arrives. Furthermore, a novel under-sampling method is designed for handling the dynamic class-imbalance problem caused by gradually evolved classes. In comparison to existing methods, CBCE can adapt well to all three cases of class evolution (i.e., emergence, disappearance and reoccurrence of classes). Since CBCE mines a data stream in an on-line manner, it is capable of rapidly keeping up with the gradual evolution of the data stream.

Moreover, CBCE avoids maintaining a large size of base learners and makes it flexible to class evolution. Empirical studies verify the reliability of CBCE and show that it outperforms other state-of-the-art class evolution adaptation algorithms, not only in terms of the adaptation ability of various evolution scenarios but also the overall classification performance. However, CBCE still suffers from some drawbacks. For example, a disappearing class might be of less importance than non-evolved or emerging classes in some real-world applications. In such cases, since CBCE put more emphasis on evolved classes, its performance may decay on non-evolved classes. Besides, mining task for massive and complex evolved classes (e.g., minority classes with sub-concepts) is still difficult in data stream mining. A potential future work would be to expand CBCE to overcome these difficulties.

A. Disadvantages of Existing System

- Avoids maintaining large size of base learners.
- Disappearance class having less importance.
- Mining task for complex & massive evolved classes is still difficult.

IV. SOFTWARE REQUIREMENT SPECIFICATION

- Operating System: Windows OS/Linux
- Hardware: Core i5 & 2GB RAM
- Software: Weka Tool
- Dataset: Twitter Data Set, KDD Cup 99 Network intrusion Dataset

A. Mathematical Model

Whole System consist of S=I,P, ,P,C,R,N, Success, Failure
I=Input
I=(I1,I2,I3)
I1=User1
I2=User2
I3=User3
D=Dataset
P=Process
C=updated Class
R=Reoccurrence Class
N=Novel Class

B. Success

- Upload Data Set Successfully.
- Application Start.
- Disappearance Class removed.
- Novel & Reoccurrence classes are labeled & classified.

C. Failure

- Application not started.
- Not Found any Disappeared Class.
- Classes are not labeled properly.

V. PROPOSED SYSTEM

In proposed system the problem of class evolution is removed by adding a new model for complex & massive data sets. In existing approach mining task for complex & massive is still difficult problem. This problem is eliminated in this proposed model called CCEM (Complex Class Ensemble Model).In this model CCEM basically handles problems of Concept drift & dynamic imbalance problem like CBCE. The main advantage of this model is that it handles large & massive chunk size data streams & also emphasize on class disappearance which is not occurred in previous studies. The proposed model is shown in figure.

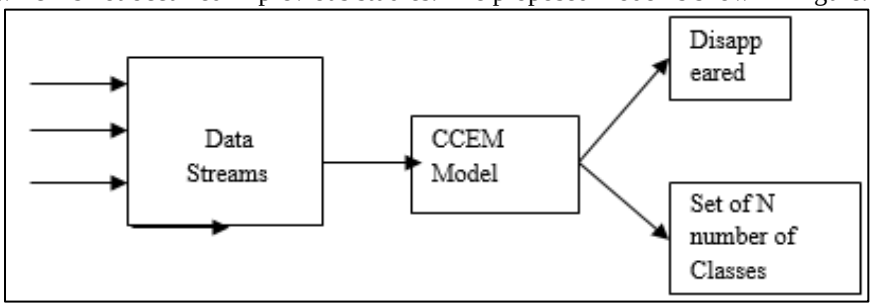

Fig. 2: Complex Class Ensemble Model (CCEM) Model

As shown in fig. Data streams from different sources are classified by CCEM model into class emergence, class reoccurrence & class disappearance. CCEM model first separate class disappearance from data streams by calculating prior probability. If it less than threshold value it is considered as class disappearance. After sorting for massive & large data sets CCEM model calculate arrival time of remaining class & assign unique ID to each class. According to arrival time of class it is classified by model. So that for large data set also model updates & gives accurate result. In comparison to existing methods, CCEM can adapt well to all three cases of class evolution (i.e., emergence, disappearance and reoccurrence of classes). CCEM mines a data stream for classes which is gradually evolved. CCEM avoids maintaining a large size of base learners and makes it flexible to class evolution. Empirical studies verify the reliability of CCEM and show that it outperforms other state-of-the-art class evolution adaptation algorithms, not only in terms of the adaptation ability of various evolution scenarios but also the overall classification performance.

In such cases, since CCEM put more emphasis on evolved classes, its performance may decay on non-evolved classes. In CCEM, when data is received hen it is divided into chunks & then predicts its label .After obtaining the true label of this example, each model is updated to track the up-to-date concept. If a novel class emerges, a new model corresponding to this class is initialized. A sufficiently small prior probability of a class implies its disappearance. In this case, the corresponding model is inactivated but still conserved. If a disappeared class reoccurs, the corresponding model will be re-activated with the prior probability of the class being re-estimated from the current data. In order to handle the dynamic class imbalance problem caused by the gradual process of class evolution, models use under-sampling with a dynamic probability to sample the examples to balance the training data. It is noted that all active models are used for classification, with decision determined by choosing a class whose model outputs the highest score. A change detection method is used to monitor changes in the class conditional probability distributions corresponding to each model. If a change is detected, the corresponding model is reset.

For example, in an early stage, an event may be discussed by a few participants on Twitter; the topic grows in popularity over a period of time and then eventually fades away from attention. Motivated by this consideration, this work investigates the class evolution problem with gradually evolved classes. Gradual evolution of classes refers to the case that classes appear or disappear in a gradual rather than transient manner, i.e., the EGR changes more smoothly. A novel class-based ensemble approach, namely Class-Based ensemble for Class Evolution (CCEM), is proposed. In contrast to the above-mentioned existing approaches, which process a data stream in a chunk-by-chunk manner and build a base learner for each chunk, CCEM maintains a base learner for every class that has ever appeared and updates the base learners whenever a new example arrives (i.e., in a one-pass manner). Furthermore, a novel under-sampling method is also designed to cope with the dynamic class-imbalance problem induced by gradual class evolution.

Class reoccurrence means that an example with the label of a disappeared class is received again. Effective handling of class reoccurrence could make use of past training efforts. For the inactivated model of a disappeared class, it can be used again for classification when an example with an old label arrives, which makes CCEM efficient. Once class reoccurrence happens, the model re-estimates the prior probability in the same way as class emergence, and activates the model in classification.

A. Comparison with Existing System

- CCEM model can maintain large size of data sets compared with existing system it cannot work on large size of chunk or dataset
- CCEM model gives more importance to class disappearance & existing system emphasize on novel & reoccurred class only.
- Mining Task for massive & complex evolved classes can be done by using model & as compared with existing system it works only on small chunk of data.

VI. PERFORMANCE MEASURES

In proposed system different performance measures are considered. In proposed model classification different tools are considered such as Weka tool. Before classification prior probability of each class is calculated & accordingly classifies & identify whether class is disappeared, novel or reoccurred. For performance measures different measures like F1 score, precision or recall can be considered in proposed system. CCEM model gives better result as compared with previously model used for classification.

VII. RESULT & CONCLUSION

In CCEM model data streams are classified into different classes of emergence, disappearance & reoccurrence. The classes which are disappeared they are removed from data sets. So it will help to reduce the size of data streams. In this model we assign a unique ID to each classes depending upon its arrival time. So proposed model classify data accordingly to its unique ID. In previous work no any work upon complex & massive data sets, So proposed model helps to maintain such a huge stream data & also increase accuracy as compared with other model in data mining.

In this we take different data sets of different data sets. As shown in figure the comparison

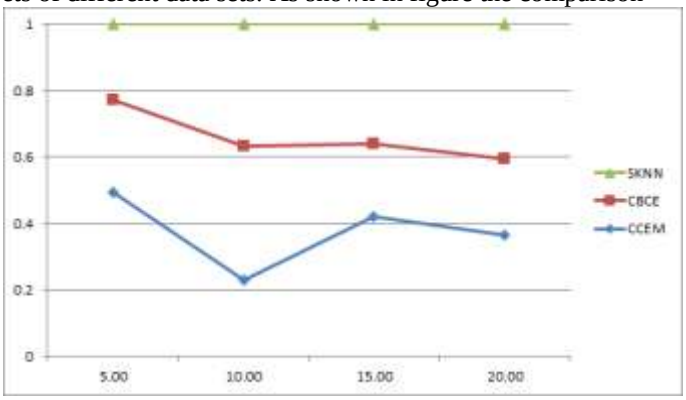

Fig. 3: Influence of Delay Factor with Different Approaches.

From the comparison of the mining results, CCEM is shown to outperform other algorithms in adapting different types of class evolutions for both the evolved classes and the whole data streams. The empirical study also confirms that CCEM has a satisfactory time efficiency in mining data streams. Generally speaking, CCEM is able to construct a satisfactory model for handling gradual class evolution. However, the results on tweet stream—20 classes also show that data stream mining with multiple and complex evolved classes is still a tough problem. To further investigate CCEM, the influence of decay factor and disappearance threshold is studied, as shown in Fig.. It can be found that a decay factor of 0.9 allows CBCE to achieve a good result in all the data streams. Considering the tracking of prior probability of classes as well, 0.9 is recommended as the default setting of decay factor. Disappearance threshold is a parameter specific to each application. From the result, a small value (e.g.,less than 2_{16}) is a good initial setting.

ACKNOWLEDGMENTS

I take this golden opportunity to owe our deep sense of gratitude to my project guide Prof. S.V.Bodake help and valuable guidance with a lot of encouragement throughout this paper work, right from selection of topic work up to its completion. My sincere thanks to Head of the Department of Computer Engineering Prof.B.K.Sarkar who continuously motivated and guided us for completion of this paper. I am also thankful to our PG Coordinator, all teaching and nonteaching staff members, for their valuable suggestions and valuable co-operation for partially completion of this work. I specially thank to those who helped us directly-indirectly in completion of this work successfully.

REFERENCES

[1] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining data streams: A review,” SIGMOD Rec., vol. 34, no. 2, pp. 18–26, 2005.
[2] P. Domingos and G. Hulten, “Mining high-speed data streams,”inProc. 6th ACM SIGKDD Int. Conf. Know. Discovery Data Mining,2000, pp. 71–80.
[3] A. Bifet and R. Gavald_a, “Learning from time-changing data with adaptive windowing,” in Proc. SIAM Int. Conf. Data Mining, 2007,pp. 443–448.
[4] W. N. Street and Y. Kim, “A streaming ensemble algorithm (SEA) for large-scale classification,” in Proc. 7th ACM SIGKDD Int. Conf. Know. Discovery Data Mining, 2001, pp. 377–382.
[5] M. Karnick, M. Ahiskali, M. Muhlbaier, and R. Polikar, “Learning concept drift in nonstationary environments using an ensemble of classifiers based approach,” in Proc. IEEE Int. Joint Conf. Neural Netw.,Jun. 2008, pp. 3455–3462.
[6] M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham,“Classification and novel class detection in concept-drifting data streams under time constraints,” IEEE Trans. Know. Data Eng.,vol. 23, no. 6, pp. 859–874, Jun. 2011.
[7] S. Wang, L. Minku, and X. Yao, “A learning framework for online class imbalance learning,” in Proc. IEEE Symp.Comput.Intell.Ensemble Learn., Apr. 2013, pp. 36–45.
[8] N. Japkowicz, “Concept-learning in the presence of between-class and within-class imbalances,” in Proc. 14th Biennial Conf. Can. Soc.Comput. Stud. Intell.: Adv. Artif. Intell., 2001, pp. 67–77.
[9] P. Mallapragada, R. Jin, and A. Jain, “Non-parametric mixture models for clustering,” in Proc. Int. Conf. Struct., Syntactic, and Statistical Pattern Recog., 2010, vol. 6218, pp. 334–343.