MINING DATA STREAMS WITH CONCEPT DRIFT IN MASSIVE ONLINE ANALYSIS FRAME WORK

PROF. DR. P. K. SRIMANI
Former Director, R & D, Bangalore University, Bangalore, Karnataka, India

MRS. MALINI M PATIL
Associate Professor, Dept. of ISE, JSSATE,VTU, Bangalore, Karnataka, India
Research Scholar, Bharathiar University, Coimbatore, Tamilnadu, India
profsrimanipk@gmail.com, patilmalini31@gmail.com

Abstract—The advancement of the technology has resulted in the data generation with increasing rate of data distribution. The generated data is called as 'data stream'. Data streams can be mined only by using sophisticated techniques. The stream data mainly comes from mobile applications, sensor applications, network monitoring, traffic management, weblogs etc. But the concepts often change with time. Weather forecasting data is a good examples here. The model built on old data is inconsistent with the new data and regular updation of the model is necessary. This type of change in a data stream is called as concept drift. The paper aims at mining data streams with concept drift in Massive Online Analysis Frame work by using Naive Bayes algorithm using classification technique. The authors also generated their own data set generator OUR-GENERATOR for the analysis. The other generators used are LED, RANDOMRBF, WAVEFORM, SEA, STAGGER and HYPERPLANE with concept drift. Along with Our Generator the other three static generators used are: Electricity, Airline and Forest Cover. The performance of the Naive Bayes on RANDOMRBF generator is found to be excellent but equally it is best on OUR_GENERATOR also which is first of its kind in the literature.

***Keywords*—** Concept drift, Massive data mining, Data streams, Naive Bayes, Accuracy, our_generator.

1 Introduction

The present technology has contributed for the increasing rate of data generation with varying data distributions. This is mainly because of different mobile applications, sensor applications, measurements in network traffic monitoring and management, log records and click streams in search engines, web logs, emails, blogs, twitter posts etc. Thus a *data stream is defined as an ordered sequence of items that arrive in timely order* [1]. Data streams are different from traditional databases. They are continuous, unbounded, usually come in high speed and have a data distribution which often changes with time [2]. Mining a stream data is referred to as data stream mining or Massive Data Mining (MDM).

Few important features of data streams are:(i) data streams are huge in size. (ii)data streams are continuous in nature. (iii)data streams are fast changing and require fast response. (iv) random access of data is not possible. (vi) storage of data streams is limited, only the summary of the data can be stored. (viii) mining such data needs sophisticated techniques.

The main requirements in mining data streams are summarized as follows: (i) the example has to be processed at a time, and inspected only once. (ii) limited amount of memory can be used. (iii) it should work in a limited amount of time. (iv) predictions can be made at any time.

Few important challenges of data streams are summarized as follows:(i) since the data collected is huge, multiple scans are not possible in data stream mining as compared with traditional data mining algorithms. (ii)the mining method of data streams should handle the change in data distribution. (iii) in the case of online data streams mining methods should be more faster than the speed of incoming data. (iv) memory management issues related to data storage and CPU speed also matter more in data stream mining.

Data streams can be classified into two types viz., static streams and evolving streams. Static streams are characterized by regular bulk arrivals. e.g. web logs, and queries on data warehouses. Evolving data streams [2] are characterized by real time updated data that come one by one in time. e.g., frequency estimation of internet packet streams, stock market data, and sensor data. Yet another important feature is that bulk data processing is not possible in evolving data streams where as it is possible in static data streams. The paper is organized as follows: Section II focuses mainly on the related work in the area of concept drift; Section III discusses about the methodology used in mining data streams with concept drift in Massive Data Mining; Section IV deals with the results and analysis respectively; Conclusions and Future Works are briefed at the end of the paper.

1.1 Concept Drift

Concept drift is best understood by weather forecasting data. When the example is keenly observed it is found that the cause of the change is hidden. Changes in the hidden context are directly proportional to changes in the target concept. It is assumed to be unpredictable. An effective learner should be designed in such a way that it should be able to track such changes and should also adapt quickly to such changes. A difficult problem in handling concept drift is differentiation of true concept drift and noise. Thus the ideal concept drift handling system should be able to adapt quickly to concept drift, robust to noise and differentiate it with noise.

The term *concept drift* [3, 4] is defined as an unforeseen substitution of one data source S1 having probability distribution P (S1), with another source S2 having probability distribution P (S2). If a data stream of length 't' has just two data generating sources S1 and S2, the number of possible change patterns is 2^t. Since the data sources are unbounded, the number of source distribution changes can be infinite. From [20] it is found that the concept drift may occur in three ways. They are:

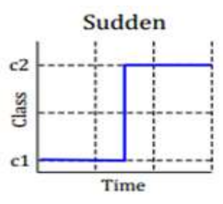
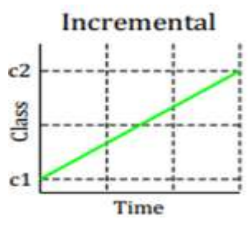
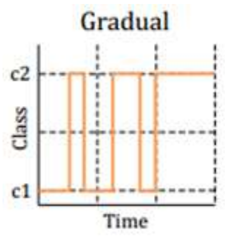
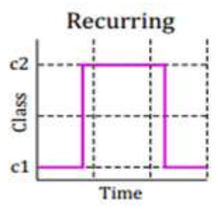
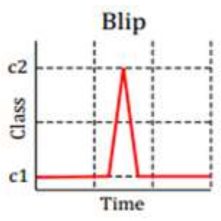
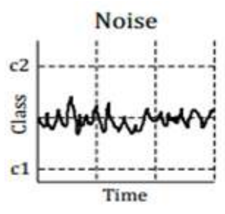
- Prior probabilities of classes, P (c_1),….,P(c_k) may change over time.
- class-conditional probability distributions, P$(X|c_i)$, i = 1,….,k might change.
- Posterior probabilities P$(c_i|X)$, i = 1,….k might change.

Thus the paper aims at mining data streams with concept drift in Massive Online Analysis Frame work by using Naive Bayes algorithm using classification technique.

1.2 Types of Concept Drifts

The different types of concept drifts available in the literature survey are listed in Table 1 along with their brief definitions.

Table 1: Types of Concept drifts definitions

Type of Concept Drift with Figure	Definition
	This type of concept drift refers to abrupt changes that instantly and irreversibly change the variables class assignment. e.g., seasonal changes on sales.
	This type of drift occurs when variables slowly change their values over time. e.g., price growth due to inflation.
	This type of drift occurs when variables slowly change their class distribution over time. e.g., changing definitions of spams.
	This type of drift represents changes that are occurring are only temporary in nature and are reverted after some time. It is also referred as local drift.
	Blip is a rare event in streaming and can be ignored as the change represented is random in nature and can be regarded as outliers. e.g., fraudulent card detection and network intrusion.
	Noise is not considered as a concept drift as it is an insignificant fluctuation which is not connected with any change in the source distribution.

2 Related Works

Literature survey reveals that the very first systems capable of handling concept drift were [17]. One of the earliest systems capable of handling concept drift is represented by STAGGER [17]. It is one of the most popular *benchmark data* for testing concept drift. It includes three simple Boolean concepts of three features with three values each. Many learning algorithms were used for base models in systems handling concept drift. These include rule-based learning, decision trees including their incremental versions, Naïve Bayes, Radial Basis Functions networks, and instance-based learning.

Conceptual clustering [13] identifies stable hidden contexts by clustering the instances assuming that similarity of context is reflected by the degree to which instances are well classified by the same concept. Another important work is presented in [12] which mainly discusses about a case based approach to spam filtering. A model is constructed then on the identified clusters. Many learning algorithms were used for base models in systems handling concept drift. These include rule-based learning [16] Decision trees, including their incremental versions [13,14,15]. In [23] the authors have characterized adaptive learning process for handling concept drift, discuss evaluation methodology of adaptive algorithms. The survey covers the different aspects of concept drift in an integrated way to reflect on the existing scattered state-of-the-art.

In [24] the authors describe two ensemble-based approaches for learning concept drift from imbalanced data. To deal with the problem of learning when the distribution generating the data changes over time, dynamic weighted majority was proposed as an ensemble method for concept drift in [25]. Yet another work is found in [26] in which a procedure based on obtaining statistics from loss of distribution by reusing the data multiple times through re-sampling.The authors developed perceptron learning model [8] using a neural network approach. Study of classification algorithms using Bayesian approach is presented in [9]. Authors have carried out both the works in massive online analysis framework (MOA). The present work is about mining data streams with concept drift using MOA frame work. The special feature of the present investigation is the generation of the data stream *'our_generator'*, which is first of its kind in the literature.

3 Methodology

The different steps of methodology of mining data streams using MOA framework are presented in this section which covers brief introduction of MOA framework, evaluation process in MOA, performance evaluators in MOA, about the availability of data sources for carrying out the experiments.

3.1 MOA framework

It is a software environment for implementing algorithms and running experiments for online learning from evolving data streams. MOA [10,11] is designed in such a way that it can handle the challenging problem of scaling up the implementation of state of the art algorithms to real world data sets. It consists of offline and online algorithms for classification and clustering. It also consists of tools for evaluation. Thus MOA is an open source frame work to handle massive, potentially infinite, evolving data streams. MOA mainly permits the evaluation of data stream learning algorithms on large streams under explicit memory limits. The method MDM mainly consists of four steps.viz. i) Select the task. ii) Select the Learner. iii) Select the Stream Generator. iv) Select the Evaluator.

3.2 Evaluation process in MOA

There are two options in the case of evaluation process in MOA[10,11]. Viz., Hold_out and Prequential.

Hold_out: When traditional batch learning reaches a scale where cross validation is too time consuming, it is often accepted to instead measure performance on a single holdout set. This is most useful when the division between train and test sets have been pre-defined, so that results from different studies can be directly compared. Hold_out evaluation gives a more accurate estimation of the accuracy of the classifier on more recent data. However, it requires recent test data that it is difficult to obtain for real datasets.

Interleaved Test-Then-Train or Prequential: Each individual example can be used to test the model before it is used for training, and from this the accuracy can be incrementally updated. When intentionally performed in this order, the model is always being tested on examples it has not seen. This scheme has the advantage that no holdout set is needed for testing, making maximum use of the available data. It also ensures a smooth plot of accuracy over time, as each individual example will become increasingly less significant to the overall average.

3.3 Performance Evaluators

MOA basically uses four different types of performance evaluator's viz., Windows Classification performance evaluator (WCPE), Basic Classification performance evaluator (BCPE), Fading Factor Classification performance evaluator (FFCPE), EWMA Classification performance evaluator. The present work mainly uses windows classification performance evaluator.

3.4 Data Sources

From the literature survey, it is found that there is a shortage of suitable and publicly available real world bench mark data sets. The UCI Machine Learning repository [6] consists of most common benchmarks for machine learning algorithms. Most of the common bench mark data sets are not suitable for data stream mining. Even the existence of concept drift is also not found in them. That's why it has become a common practice by researchers to publish results based on synthetic data sets. The basic advantage of synthetic data is, it is easier to reproduce and there is little cost in terms of storage and transmission. The present work uses three different sets of evolving and static data set generators which are available in MOA framework. The authors have developed their own data set generator *'our_generator'*. Which is a special feature of the present investigation.

3.5 Data stream generators used in the analysis

The present investigation uses 10 data set generators which are both evolving and static in nature. The description of all the data stream generators is presented briefly as shown below.

The first set consist of three types of evolving data stream generators are *LED Generator with drift* which is of type sudden and gradual concept drift, *Random RBF Generator with drift* which is of type gradual drift, *Wave Form generator with drift* is also gradual concept drift. All these are generated during configuring the experimental set up in MOA. The second set also consists of three types of evolving data stream generators are *SEA Generator, STAGGER Generator, Hyper plane Generator* for which the concept drift is added by using *concept _drift_real_stream* option of MOA framework. This particular option adds concept drift to examples in a stream. These are of type sudden and gradual concept drift.The third set consists of the real world data sets which are also publicly available. They are *Forest Cover type, Electricity, airlines* data set. These are static streams. In this case also the concept drift is added by using *concept_drift_real_stream* option of MOA framework.

The authors have developed their own data set generator named as Our_generator for the analysis purpose which is a special feature of present work. The concept drift is added by using *concept_drift_real_stream* option of MOA framework.

LED Generator with drift [11]**:** Generates a problem of predicting the digit displayed on a 7-segment LED display with drift. The goal is to predict the digit displayed on a seven-segment LED display, where each attribute has a 10% chance of being inverted. It has an optimal Bayes classification rate of 74%. The particular configuration of the generator used for experiments (led) produces 24 binary attributes, 17 of which are irrelevant. This generator is used to acquire 1,000,000 examples with sudden and gradual concept drift.

Random RBF Generator with drift [11]**:** Generates a random radial basis function stream with gradual drift. Drift is introduced by moving the centroids with constant speed. The RBF (Radial Basis Function) generator works as follows: A fixed number of random centroids are generated. Each center has a random position, a single standard deviation, class label and weight. New examples are generated by selecting a center at random, taking weights into consideration so that centers with higher weight are more likely to be chosen.

Wave Form generator with drift [11]**:** Waveform Generator Generates a problem of predicting one of three waveform types with gradual drift. It shares its origins with LED, and was also donated by David Aha to the UCI repository. The goal of the task is to differentiate between three different classes of waveform, each of which is generated from a combination of two or three base waves. The optimal Baye's classification rate is known to be 86%. There are two versions of the problem, wave21 which introduces an additional 19 irrelevant attributes.

SEA Generator [17]**:** A streaming ensemble algorithm (SEA) is used for large-scale classification. Generates SEA concepts functions. This dataset contains abrupt concept drift. It is generated using three attributes, where only the two first attributes are relevant. All three attributes have values between 0 and 10. The points of the dataset are divided into 4 blocks with different concepts. In each block, the classification is done using $f_1 + f_2 \leq \varepsilon$, where f_1 and f_2 represent the first two attributes and ε is a threshold value. The most frequent values are 9, 8, 7 and 9.5 for the data blocks.

STAGGER Generator [16]**:** Generates STAGGER Concept functions. The function uses the incremental learning method from noisy data. The STAGGER Concepts are Boolean functions of three attributes encoding objects: size (small, medium, and large), shape (circle, triangle, and rectangle), and colour (red, blue, and green). A concept description covering either green rectangles or red triangles is represented by (shape= rectangle and colour=green) or (shape=triangle and colour=red).

Hyperplane Generator [11]: Generates a problem of predicting class of a rotating hyperplane. Hyper planes are useful for simulating time-changing concepts, because we can change the orientation and position of the hyper plane in a smooth manner by changing the relative size of the weights. Concept drift is introduced to this dataset adding drift to each weight attribute $w_i = w_i + d\sigma$, where σ is the probability that the direction of change is reversed and d is the change applied to every example.

The Electricity dataset [22]: This is one of the popular benchmark data set for testing adaptive classifiers1. It has been used in over 40 concept drift experiments. The dataset covers a period of two years (45312 instances recorded every half an hour with 6 input variables). A binary classification task is to predict a rise (UP) or a fall (DOWN) in the electricity price in New South Wales (Australia). The prior probability of DOWN is 58%. The data is subject to concept drift due to changing consumption habits, unexpected events and seasonality. This dataset has an important property not to be ignored when evaluating concept drift adaptation. Suppose we employ a naive predictor that predicts the next label to be the same as the current label (the moving average of one). For instance, if the price goes UP now, it predicts that the next time step the price will go UP as well. If the data was distributed independently, such a predictor would achieve 51% accuracy. However, if we test this naive approach on the Electricity dataset it gives much higher 85% accuracy. This happens because the labels are not independent; there are long consecutive periods of UP and long consecutive periods of DOWN.

The Airline data [22]**:** This data set consists of flight arrival and departure details for all commercial flights from 1987 to 2008 within USA. This is a large dataset with nearly 120 million records (11.5 GB memory size). The best set of queries with respect to this data set are: When is the best time of day/day of week/time of year to fly to minimise delays? Do older planes suffer more delays? How does the number of people flying between different locations change over time? How well does weather predict plane delays? Can you detect cascading failures as delays in one airport create delays in others? Are there critical links in the system?. The dataset was cleaned and records were sorted according to the arrival/departure date (year, month, and day) and time of flight. Its final size is around 116 million records and 5.76 GB of memory. There are 13 attributes, each represented in a separate column: Year, Month, Day of Month, Day of Week, CRS Departure Time, CRS Arrival Time, Unique Carrier, Flight Number, Actual Elapsed Time, Origin, Destination, Distance, and Diverted. The target variable is the Arrival Delay, given in seconds.

The Forest cover [7]: This type of data set is related to US Forest Servic(USFS). The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS)-Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS). Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types). Number of instances (observations) is 581012, Number of Attributes is 54, and missing attribute values are none.

Our_Generator: Generates the data streams using the random number function. The data generation method is based on the customer buying pattern in the market basket data. The number of items is assumed as 26. The data is outputted in the form of flat file which mainly includes maximum column width (≥ 5 and $\leq$ 26) minimum column width (assumed as 4 always). Number of items is constant (26). The total number of items in one transaction varies from minimum column width to maximum column width. Using the function {elements_count = random () % maxi_trans + mini_trans} the required number of transactions are generated. Once the data stream is generated the concept drift is introduced artificially by adding noise (40%) for evaluation purpose. An effective code is designed in order to generate the data stream of 'n' number of instances, where n=200,000 in the present work.

3.6 Algorithm used in the analysis

The present work uses Naïve Bayes algorithm which performs classic bayesian prediction while making naive assumption that all inputs are independent. It is known for its simplicity and low computational cost. In [22] the authors have explained the steps of Naïve Bayes algorithm and are presented as follows:

1. Let D be a training set of tuple and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $\mathbf{X} = (x_1, x_2,\dots,x_n)$, depicting 'n' measurements made on the tuple from n attributes, respectively, $A_1, A_2, \dots, A_n$.

2. Suppose that there are m classes, $C_1, C_2,\dots,C_m$. Given a tuple, **X**, the classifier will predict that **X** belongs to the class having the highest posterior probability, conditioned on **X**. That is, the naïve Bayesian classifier predicts that tuple **X** belongs to the class C_i if and only if

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \le j \le m;\ j \ne i \qquad (1)$$

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|X) = P(X|C_i)\, P(C_i) / P(X) \qquad (2)$$

3. As $P(\mathbf{X})$ is constant for all classes, only $P(\mathbf{X}|C_i)\, P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = , , , = P(C_m)$, and therefore maximize $P(\mathbf{X}|C_i)$; Otherwise maximize $P(\mathbf{X}|C_i)P(C_i)$. It is also noted that the class prior probabilities may be estimated by $P(C_i) = |C_{i,\ D}/D|$, where $|C_i, D|$ is the number of training tuples of class C_i in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(\mathbf{X}|C_i)$. In order to reduce computation in evaluating $P(\mathbf{X}|C_i)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \qquad (3)$$

We can easily estimate the probabilities $P(x_1|C_i)$, $P(x_2|C_i)$...$P(x_n|C_i)$ from the training tuples. x_k refers to the value of attribute A_k for tuple **X**.

5. In order to predict the class label of **X**, $P(\mathbf{X}|C_i)\, P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple **X** is the class C_i if and only if

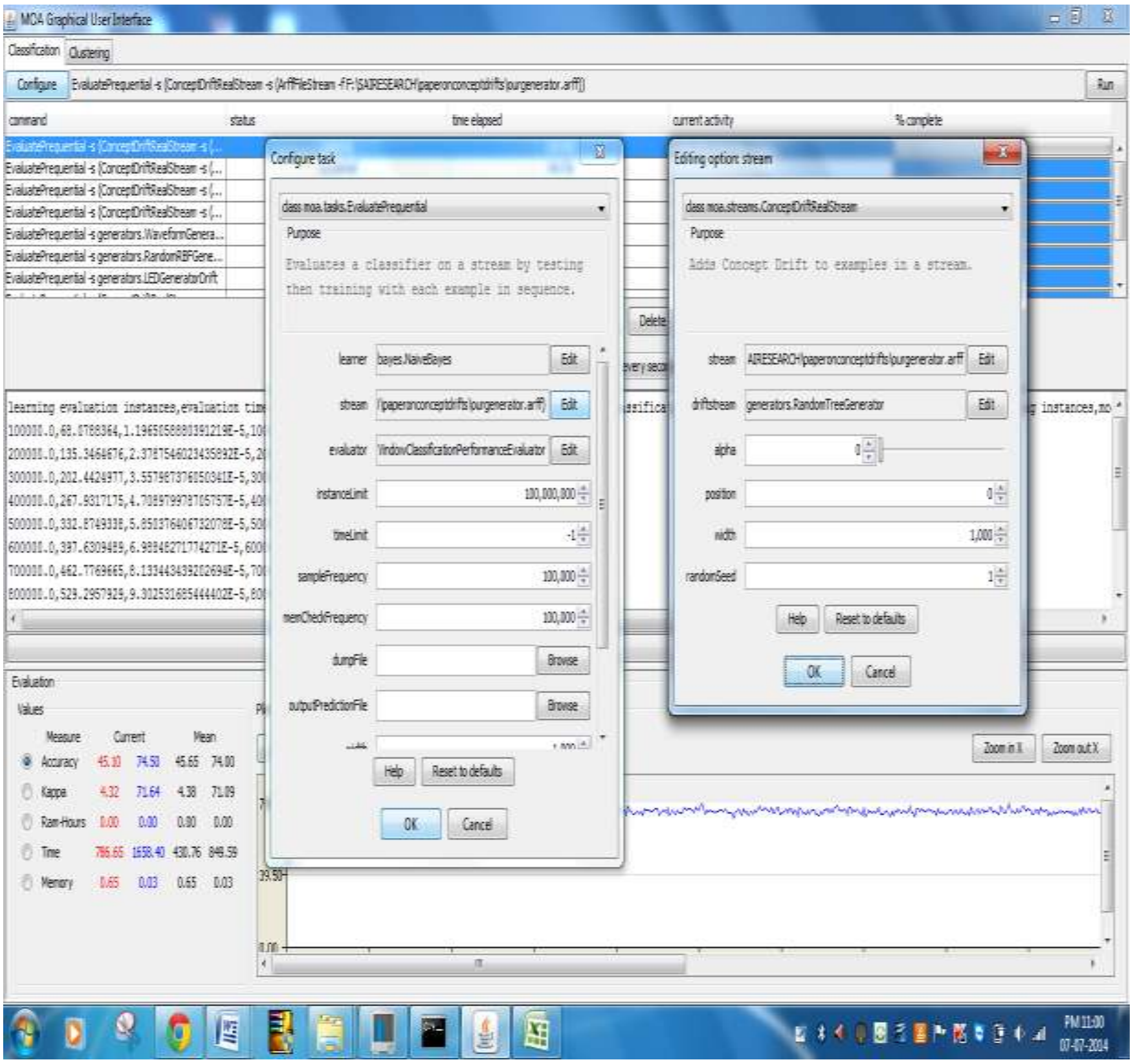
$$P(X|C_i)P(C_i) > P(X|C_j)\, P(C_j) \text{ for } 1 \le j \le m;\ j \ne i \qquad (4)$$

In other words, the predicted class label is the class C_i for which $P(X|C_i)P(Ci)$ is the maximum. In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers. However, in practice this is not always the case, owing to inaccuracies in the assumptions made for its use, such as class conditional independence, and the lack of available probability data. Bayesian classifiers are also useful in that they provide a theoretical justification for other classifiers that do not explicitly use Bayes' theorem. For example, under certain assumptions, it can be shown that many neural network and curve-fitting algorithms output the *maximum posteriori* hypothesis, as does the naïve Bayesian classifier.

4 Experiments and Results

The experimental Work in MOA is divided into four different tasks. They include classifier training, learner evaluation, stream file generation, and stream speed measurement. Tasks can be executed from a graphical user interface, as well as from the command line. The user interface allows to run many tasks concurrently, controlling their progress and presenting partial results. It also allows to create data streams on the fly using generators, by joining several streams, or by filtering streams. The framework also provides an interesting feature that allows to add concept drift to stationary data streams. The snapshot of MOA configuration task is shown in figure 1.

The present experimental set up consists of following parameters. Instance Limit: 100,000,000, Performance Evaluator: Windows Classification Performance Evaluator, Evaluation Task: Evaluate Prequential, Classifier: Naive bayes. The instance limit is different for static streams as they are real world data sets. They have their own predefined instance limit and it is used. Table 2, 3 and 4 presents the values of accuracy, kappa, Ram-hours, time utilized and memory used for the data stream generators considered with respect to the NB classifier with concept drift (LED, RANDOMRBF, WAVEFORM), by adding concept drift (SEA, STAGGER, HYPERPLANE) and for static streams by adding concept drift (FOREST COVER TYPE, ELECTRICITY, AIRLINES and OUR_GENERATOR).



```
EvaluatePrequential -s (ConceptDriftRealStream -s (ArffFileStream -f F:\SAIRESEARCH\paperonconceptdrifts\purgenerator.arff))
```

```
command                                          status    time elapsed    current activity    % complete
EvaluatePrequential -s (ConceptDriftRealStream -s (...
EvaluatePrequential -s (ConceptDriftRealStream -s (...
EvaluatePrequential -s (ConceptDriftRealStream -s (...
EvaluatePrequential -s (ConceptDriftRealStream -s (...
EvaluatePrequential -s generators.WaveformGenera...
EvaluatePrequential -s generators.RandomRBFGene...
EvaluatePrequential -s generators.LEDGeneratorDrift
```

```
learning evaluation instances,evaluation time
100000.0,68.0788364,1.1965058880391219E-5,10
200000.0,135.3464676,2.3787546023435892E-5,2
300000.0,202.4424977,3.5579873760503415E-5,30
400000.0,267.9317175,4.7085799787057578E-5,40
500000.0,332.8749338,5.8503764067320785E-5,50
600000.0,397.6309489,6.9884627177427IE-5,600
700000.0,462.7769665,8.1334434392026948E-5,70
800000.0,529.2957929,9.3025316854444402E-5,80
```

Configure task
class moa.tasks.EvaluatePrequential
Purpose
Evaluates a classifier on a stream by testing then training with each example in sequence.

learner	bayes.NaiveBayes
stream	\paperonconceptdrifts\purgenerator.arff)
evaluator	WindowClassificationPerformanceEvaluator
instanceLimit	100,000,000
timeLimit	-1
sampleFrequency	100,000
memCheckFrequency	100,000
dumpFile	
outputPredictionFile	

Editing option: stream
class moa.streams.ConceptDriftRealStream
Purpose
Adds Concept Drift to examples in a stream.

stream	AIRESEARCH\paperonconceptdrifts\purgenerator.arff
driftstream	generators.RandomTreeGenerator
alpha	0
position	0
width	1,000
randomSeed	1

Measure	Current		Mean	
Accuracy	45.10	74.50	45.65	74.00
Kappa	4.32	71.64	4.38	71.09
Ram-Hours	0.00	0.00	0.00	0.00
Time	786.65	1658.40	430.76	849.59
Memory	0.65	0.03	0.65	0.03

PM 11:00
07-07-2014

Fig. 1. Configuration model in MOA

Table 2. Results of evaluation measures for NB for evolving streams by adding concept drift

	SEA	STAGGER	HYPER PLANE
ACCURACY (%)	73.62	65.24	74.00
KAPPA	44.43	33.92	71.09
RAM HOURS	0.00	0.00	0.00
TIME(sec)	306.79	553.16	849.59
MEMORY(Mb)	0.01	0.01	0.03

Table 3. Results of evaluation measures for NB for evolving streams with concept drift

	LED	WAVE FORM	RANDOM RBF
ACCURACY (%)	72.00	73.49	80.51
KAPPA	43.96	43.44	70.74
RAM HOURS	0.00	0.00	0.00
TIME(sec)	622.77	485.49	1307.45
MEMORY(Mb)	0.01	0.08	0.01

Table 4. Results of evaluation measures for NB for static streams with concept drift

	AIR LINE	COVER TYPE	ELECTRI CITY	OUR_GEN ERATOR
ACCURACY (%)	60.52	73.39	45.45	74
KAPPA	26.90	46.10	40.31	71.09
RAM HOURS	0.00	0.00	0.00	0.00
TIME(sec)	1265.8	467.86	462.32	849.59
MEMORY (Mb)	0.05	0.01	0.05	0.03

From table 2 it is evident that the performance of NB algorithm is good on HYPERPLANE generator with accuracy= 74%, Kappa = 71.09 when compared to others. It is to be noted that the concept drift is added to all these generators i.e. SEA, HYPERPLANE, STAGGER by using concept drift real stream from the massive online analysis framework. The distinctive features of the constancy of ram-hours and memory usage are observed in the present investigation.

From table 3 it is evident that the performance of NB algorithm is good on RANDOMRBF generator with accuracy = 80.51%, Kappa = 70.74 when compared to others. It is to be noted that LED, WAVEFORM and RANDOMRBF generators are used in the analysis are evolving streams with concept drift in MOA framework. The distinctive features of the constancy of ram-hours and memory usage are observed in the present investigation.

From table 4 it is evident that the performance of NB algorithm is good on OUR generator with accuracy = 74%, Kappa = 71.09 when compared to others. It is to be noted that AIRLINE, COVERTYPR, ELECTRICITY, OUR GENERATOR are static streams and the concept drift is added by using concept drift real stream from the massive online analysis framework. The distinctive features of the constancy of ram-hours and memory usage are observed in the present investigation. Figures 2, 3 and 4 represent the different graphs generated for the results obtained and are self explanatory.

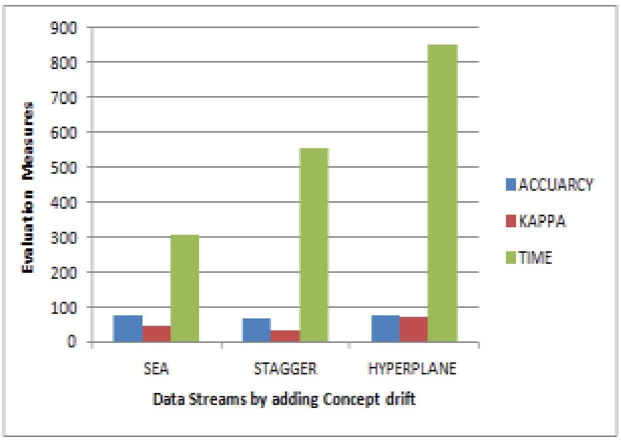


Figure 2. Graph of Evaluation Measures (Accuracy, Kappa and Time) Vs Evolving Data Streams for NB.

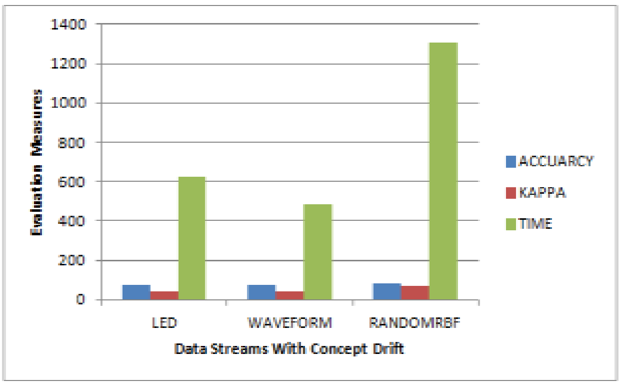


Figure 3. Graph of Evaluation Measures (ACCU,Kappa and Time) Vs Evolving Data Streams for NB algorithm

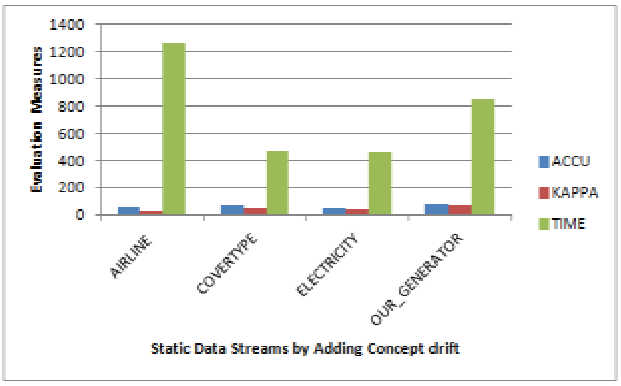


Figure 4. Graph of Evaluation Measures (ACC,Kappa and Time)Vs Static Data Streams for NB algorithm.

5 Conclusion

The present study reveals the following conclusions and contributions.

- In the present investigation the authors have used ten data set generators of which 6 are evolving streams and 4 are static streams to perform massive data mining with concept drift using massive online analysis frame work.
- The learning model uses Baysian approach
- SEA, STAGGER and HYPERPLANE are evolving data streams for which concept drift is added using concept_drift_real_stream. These are of type sudden and gradual concept drift types.
- RANDOMRBF, LED, WAVEFORM, are other three evolving data set generators which are generated with concept drift during running the experimental set up in MOA frame work.
- AIRLINE, COVERTYPR, ELECTRICITY, OUR GENERATOR are static streams and the concept drift is added by using concept_drift _real_stream from the massive online analysis framework.
- The key feature of present analysis is that a code is designed to generate the data stream OUR_GENERATOR and the experiment is performed on this generator also.
- The performance of NB algorithm is good on HYPERPLANE generator with accuracy= 74%, Kappa = 71.09 when compared to others.
- NB algorithm is good on RANDOMRBF generator with accuracy = 80.51%, Kappa = 70.74 when compared to others.
- It is evident that the performance of NB algorithm is good on OUR generator with accuracy=74%, Kappa = 71.09 when compared to others.
- Interesting point is the memory consumed remains same in all the three data streams. RAM hours taken by the NB is same in all the data stream generators.
- The distinctive features of the constancy of ram-hours and memory usage are observed in the present investigation.
- This is first of its kind in the literature because no generated data stream is available in the data repository other then the built in data streams available in the MOA frame work.
- The future work includes the extension of the present investigation using hoefding tree approach in MOA framework.

The main contributions of the paper are as follows:

- The paper handles concept drift in both evolving and static streams in MOA frame work.
- The classification method using bayesian approach is used for handling concept drifts**.**
- The data set generator contributed by the authors i.e., Our_generator also performs equally well with respect to the other data set generators present in MOA.
- Different variations of noise levels are considered here as concept drifts, which are added during the configuration of the model.
- From the results it is to be noted that, the important challenges of data streams such as usage of limited amount of memory, ram hours and less time are achieved to the maximum extent.

6 Acknowledgements

One of the authors Mrs. Malini M. Patil acknowledges JSSMVP's, J.S.S Academy of Technical Education, Bangalore, Karnataka and Bhartiyaar University, Coimbatore, Tamilnadu, India for providing the facilities for carrying out the research work.

References

[1] Charu Aggarwal, Data Streams: Models and Algorithms. Series: *Advances in Database Systems, Vol. 31, XVIII, 354 p, ebook ,Springer, Berlin Heidelberg.*

[2] Mohamed Medhat Gaber, Arkady Zaslavsky and Shonali Krishnamurthy, Data Streams: Models and Methods, Vol 31, pp., 39-59, *Book, Springer., Berlin Heidelberg.*

[3] Jing Gao,Wei Fan, Jiawei Han and Philip S. Yu., ,General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions, *SIAM International Conference on Data Mining*, Minneapolis, 2007, 3-14.

[4] Dariusz Brzesinski, Mining Data streams with concept drift, *Ph. D thesis*, 2010.

[5] Albert Bifet, Eibe Frank, Geoffrey Holmes, Bernard Pfahringer, Accurate Ensembles for Data Streams Combining Restricted Hoeffding Trees Using Stacking, *Journal of Machine Learning Research*, 225-240,2007

[6] David Aha, *UCI Machine Learning Repository*, 2007.

[7] Srimani P.K. and Malini M. Patil, Simple perceptron model (SPM) for evolving streams in massive data mining (MDM), *International Journal of Neural Networks.* 1(2):20-24, 2012.

[8] Srimani P.K. and Malini M. Patil, 2012. Massive data mining on Data streams Using Classification Algorithms. *International Journal of Engineering Science and Technology*, 4(6):2839-2848.

[9] Albert Bifet, Geoff Holmes, Richard Kirby and Bernard Pfahringer, 2010, MOA: Massive Online Analysis,*Journal Machine learning Research*, pp:1601-1604.

[10] Albert Bifet and Richard Kirkby, 2009, Data stream mining: A practical approach. *Technical report,*The University of Waikato. Newzealand.

[11] Cunningham P., Nowlan N., Delany S.J. and Haahr M.,2003. A Case-Based approach to spam filtering that can track concept drift. In the *Proceedings of ICCBR-2003 Workshop on Long-Lived CBR Systems.*

[12] Hulten G., Spencer L. and Domingos P., 2001. Mining time-changing data streams. In the Proceedings of 7th ACM SIGKDD, *International Conference on Knowledge Discovery and Data Mining* ,KDD-ACM Press, pp:97-106.

[13] Kolter J.Z., Maloof M.A., 2003. Dynamic weighted majority: A new ensemble method for tracking concept drift, 3rd *IEEE International Conference on Data Mining ICDM-2003*, IEEE CS Press, pp:123- 130.

[14] Kubat M., Widmer G.,1994. Adapting to drift in continuous domains,Technical Report, *Austrian Research Institute for Artificial Intelligence,* Vienna.

[15] Schlimmer J.C., Granger R.H. ,1986. Incremental learning from noisy data. *Journal of Machine Machine Learning*, 1(3): 317-354.

[16] Street W., Kim Y. ,2001. A Streaming Ensemble Algorithm(SEA) for large-scale classification. In the *Proceedings of 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp: 377-382.

[17] Wang H., Fan W., Yu P.S., Han J., 2003. Mining concept-drifting data streams using ensemble classifiers, In proceedings *.9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining KDD-2003*, ACM Press, pp:226-235.

[18] Blackard, Jock A. and Denis J. Dean. 2000. Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables. Journal of *Computers and Electronics in Agriculture* 24(3):131-151.

[19] Jiawei Han and Micheline Kamber. 2006. *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, Sanfransisco, CA.

[20] Mark G.Kelly, David J.Hand, and Niall M.Adams. 1999, The impact of changing populations on classifier perfromance. *Proceedings of International Conference on knowledge discovery in data bases*. pp:367-371.

[21] www.moa.cms.waikato.ac.nz

[22] Joao gama˜ , Indre˙ zliobait, Albert bifet, Mykola pechenizkiy, Abdelhamid bouchachia, A Survey on Concept Drift Adaptation, 2013, *ACM Computing Surveys*, Vol. 1, No. 1, Article 1.

[23] Gregory Ditzler, Robi Polikar, Incremental Learning of Concept Drift from Streaming Imbalanced Data, 2013, *IEEE Transactions on Knowledge & Data Engineering*, vol.25, no. 10, pp. 2283-2301.

[24] Dhouha Mejri, Riadh Khanchel and Mohamed Limam, An ensemble method for concept drift in non-stationary environment, 2013, *Journal of Statistical Computation and Simulation*, vol 83, No 6, pages 1115-1128.

[25] Maayan Harel, Koby Crammer, Ran El-Yaniv , Shie Mannor, 2014 Concept Drift Detection Through Resampling, Proceedings of the 31 st *International Conference on Machine ,Learning*, Beijing, China, 2014. JMLR: W&CP volume 32.