Arab J Sci Eng (2017) 42:705–714
DOI 10.1007/s13369-016-2317-x

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

RGNBC: Rough Gaussian Naïve Bayes Classifier for Data Stream Classification with Recurring Concept Drift

D. Kishore Babu[1] · **Y. Ramadevi**[2] · **K. V. Ramana**[3]

Received: 1 February 2016 / Accepted: 20 September 2016 / Published online: 29 September 2016

Abstract Due to the necessity of performing classification in streaming environments, researchers have developed various stream classification methods by handling concept drift. But, recurring concept drift is a challenging problem in data stream as the dimension of the data is not static over the period of time. By considering the recurring concept drift, this paper proposes a new classifier model, called Rough Gaussian Naïve Bayes Classifier (RGNBC) for the data stream classification. Here, two new contributions are made to handle the challenges of recurring concept drift. The first contribution is to utilize the rough set theory for detecting the concept drift. Then, gaussian naïve classifier is modified mathematically to handle the dynamic data without using the historic data. Also, the classification is performed using the posterior probability and the objective function which considers the multiple criteria. The proposed RGNBC model is experimented with two large datasets, and the results are validated against the existing MReC-DFS algorithm using sensitivity, specificity and accuracy. From the results, we proved that the proposed RGNBC model obtained the maximum accuracy of 74.5 % while compared with the existing algorithm.

Keywords Data stream · Recurring concept drift · Naïve bayes · Rough set theory · Classification

✉ D. Kishore Babu
kishorebabujntu@gmail.com

[1] Jawaharlal Nehru Technological University, Kakinada, India

[2] CBIT, Gandipet, Telangana, India

[3] JNTU, Kakinada, Andhra Pradesh, India

1 Introduction

Information extraction as models and patterns from continuous data streams is referred as data stream mining [1–3]. For data stream mining, literature presents several research works [3–5] which are mainly dedicated only to static environments by reading the complete dataset for the mining process. These datasets are stored electronically and can be accessed whenever it is required by the mining algorithm. Furthermore, the target concepts should be learned by the classifier which is a kind of mining algorithm. In past years, several solutions for static classification were developed and several accurate classifiers are also available in the literature [6,7]. But, in latest applications, the learning algorithms are applied to dynamic environments, in which the data are generated continuously. Examples of such applications include telecommunication, sensor networks, traffic management, monitoring, and web log analysis [4]. For such applications, data classification becomes a big challenge for the researchers belonging to data stream mining community.

Commonly, the data streams are of infinite size and it cannot be stored in main memory. Thus, several challenges such as storage, querying and mining required much more attention. Here, mining is mostly linked with the computational resources to examine such a huge volume of data and therefore, it is broadly studied in the literature. It is suggested that the data streams must be processed in online manner to guarantee that the results are up-to-date and that the queries can be replied in real time with less delay [1]. Usually, data elements enter the system continuously at a high rate. Also, the concept of data can change at any time, called as concept drift [2,3,8]. Several approaches are proposed in the literature to deal with the concept drift [9–13] in data stream classification.

Although many research works [9–13] are going on to solve the problem of concept drift, some other challenges that require attention are recurring changes in concept, integration of context information and feature space evolution. For example, in feature space evolution, the set of features and their significance to the target concept may change [14]. While using the existing methods to solve these problems, learning model must be executed from the scratch but it takes more processing time [15–17]. Also, while dealing the recurring changes in concept, the previous techniques in the literature have to relearn them if the concepts are new, and not recurring [5]. However, latest approaches proposed in [18–21] deals the recurrent concept change without relearning mechanism but a main issue with the existing techniques is setting the user-defined parameters to conclude whether a current concept matches with the previous one [21].

In this paper, a new classifier called, RGNB Classifier (Rough Gaussian Naïve Bayes) is proposed for classification of evolving data streams by automatically detecting the recurring concept drift. Here, rough set theory and naïve bayes classifier are effectively combined to handle the concept drift and recurring concept change. The concept drift is automatically detected using accuracy of approximation which is computed based on lower and upper approximation by rough set theory. Then, recurring concept drift is handled by dynamically selecting the important features using feature evaluation function. The classification is performed using naïve bayes classifier which dynamically updates their probability of information based on the new data stream. The updating of probability of information is newly devised, and the final classification is done based on the dynamic weighting mechanism which is newly devised by considering the multiple objectives like, sensitivity, specificity, accuracy and error values.

The paper is organized as follows: Sect. 2 presents the review of recent works, and Sect. 3 presents the motivation behind the approach. Section 4 discusses the proposed method of rough gaussian naïve bayes classifier for data stream classification with recurring concept drift. Section 5 visualizes the comparative analysis, and conclusion is given in Sect. 6.

2 Literature Review

Table 1 presents the review of the recent works available in the literature to perform data stream classification. The most of recent works concentrate on handling concept drift problem in data stream classification [18–20] using meta classifiers. Two recent works given in [5,21] utilized the recurring concept drift to perform data stream classification. Even though many methods are available in the literature [18–21] for data stream classification, the major problems like, outlier data points, weight updating without using historic data, estimation of the samples in arrival data can be significant problem still to be addressed along with the recurring concept drift. Outlier is an observation point that is distant from other observations. Commonly, classification algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of learning algorithms and statistical methods resulting in longer training times, less accurate models and ultimately poorer results.

2.1 Challenges

The important challenges identified for classification of data streams from the literature [4,5,9,10,22,23] are explained as follows.

The dynamic building of learning model is required for adapting the classifier based on the data stream which evolved or changed every time.

Updating of model should not consider the multiple scan over the original databases because the storing of historic data is practically impossible.

Considering the concept drift in the classifier model is much important because the boundary of feature space will be changed continuously.

Due to recurring concept change, feature space is also increasing dynamically. So, adapting the classifier for the dynamic feature space is another challenge to be solved in the data stream classification.

The dynamic change of feature space implicitly considers the context changes and recurring concepts which are also important for devising data stream classification.

When considering the dynamic feature space of recurring concept drift, selecting and preserving of important features is also important challenge to be considered dynamically.

In [5], they utilized the Naive Bayes (NB) algorithm with ensemble weighting mechanism to handle the recurring concept drift for data stream classification. In their method, the ensemble weight mechanism considered the accuracy and error values. But, due to the dynamic nature of data, classes and data samples are not constant over the period of time. So, considering accuracy and error may affect the performance of the classification if one class attribute has bigger data samples. So, the multiple objective criteria like, sensitivity, specificity should be included to ensemble weighting.

3 Motivation Behind the Approach

3.1 Problem Statement

The problem considered here is to perform the data stream classification by considering the recurring concept drift. Let us assume that the input data stream is D which is updated

Table 1 Literature review

Authors	Contribution	Advantages	Disadvantages
Al-Khateeb et al. [18]	Class-based ensemble	Advantage of detecting novel classes more efficiently and properly distinguishing between recurring class and novel class in the presence of concept drift	Refine the base model to detect the recurring concept with abrupt concept changes
Masud et al. [19]	Auxiliary ensemble-based classifier	Utilizes much simpler operations to update a model	Outlier data points heavily affect the stream classification
Gama and Kosina [20]	Meta learning-based classifier	It provides information about the recurrence of concepts and rapidly adapts decision models when drift occurs	Require monitoring the evolution of the learning process
Sripirakas and Pears [21]	DFT and decision tree-based classifier	It is simple to detect the recurring concept drift	Computation overhead in performing the DFT operation
Brzezinski and Stefanowski [4]	Accuracy-based weighting mechanisms	Consider the periodic weighting mechanism	Adapting weight for different data space seems tough
Gomes et al. [5]	Dynamic feature space-based model learning	No holdout set is needed for testing, making use of all the available training data	Distribution of data is required to do classification
Masud et al. [22]	Concept drift and concept evolution-based ensemble classifier	Addresses four major challenges, namely, infinite length, concept drift, concept evolution, and feature evolution	It finds difficult to distinguish from the actual arrival of a novel class
Abdulsalam et al. [23]	Combines the ideas of streaming decision trees and random forests	It quickly records the new expected classification accuracy after the changes are presented in the stream	Handling multiple classes with this hybrid model is difficult

continuously for every time.

$$D = \{d_t; 1 \leq t \leq N\} \tag{1}$$

At time t, the newly coming database is denoted as d_t which have the n_t number of data objects and m_t number of features.

$$d_t = \left\{d_t^{jk}; 1 \leq j \leq n_t; 1 \leq k \leq m_t\right\} \tag{2}$$

The important problem considered here is the recurring concept drift where the dimension of the features is varied for every time. So, the attributes are varied based on the time interval t. Here, for the time period t, the number of features is m_t.

$$d_t = \left[a_t^1, a_t^2, \ldots, a_t^m\right] \tag{3}$$

Based on the recurring data stream, the classification can be performed by dynamically updating classification model as we know that the whole data cannot be utilized for the classification because the size of the data is much larger.

3.2 Contributions of the Paper

The main contributions of the paper are given as follows:

- We propose a new classifier called Rough Gaussian Naïve Bayes Classifier for Data Stream Classification with Recurring Concept Drift. Here, rough set theory in included for detecting concept drift.
- In the second contribution, gaussian naïve classifier is modified mathematically to handle the dynamic data without using the historic data. Also, the classification is performed using the posterior probability with objective function which considers the multiple criteria.

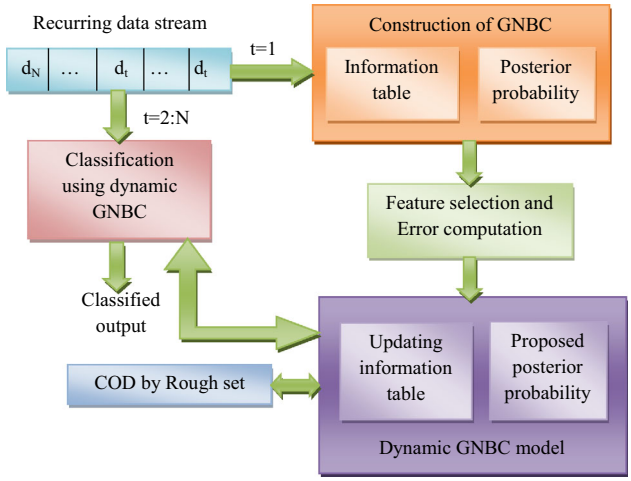

Fig. 1 Block diagram of the proposed rough gaussian naïve bayes classifier for data stream classification with recurring concept drift

4 Proposed Methodology: Rough Gaussian Naïve Bayes Classifier for Data Stream Classification with Recurring Concept Drift

This section presents a new dynamic model for data stream classification with recurring concept drift. At first, input data stream is directly read out by the proposed method at every time and the GNBC is built up initially by constructing the information table. Then, for new incoming data stream, concept of change (COD) is detected using rough set theory which have the accuracy of approximation. Once the COD is detected, the GNBC model is updated based on the new mathematical model developed in this work. This model updates the existing model based on the new data stream without storing the data tuples. Also, the proposed method utilizes the feature evaluation function and information table to handle the recurring concept drift. Figure 1 shows the block diagram of the proposed rough gaussian naïve bayes classifier for data stream classification with recurring concept drift.

4.1 Constructing Gaussian Naïve Bayes Classifier

This section presents the construction of the proposed Gaussian naïve bayes classifier of the data stream at initial stage. The Gaussian naïve bayes classifier [7] performs the classification using two important steps such as, construction of model and classification. At the model construction, the information table is constructed by including mean and variance of the every attributes. In the classification stage, the posterior probability is computed to find the class label of the input data.

Model let us assume that the input data at initial stage (d_0) is read out and it is given to construct the information table IT_t which contains two tables belonging to mean and variance of every attributes. The information is indicated as follows,

$$\mathrm{IT}_t = \left\{\mathrm{IT}_t^{\mathrm{mean}}, \mathrm{IT}_t^{\mathrm{var}}\right\} \tag{4}$$

The information table belonging to the mean $\mathrm{IT}_t^{\mathrm{mean}}$ contains a table of size $c \times m_t$ where c represents the number of classes and m_t number of attributes at time interval t.

$$\mathrm{IT}_t^{\mathrm{mean}} = \left\{\mathrm{IT}_{ak}^{\mathrm{mean}}; 1 \le a \le c; 1 \le k \le m_t\right\} \tag{5}$$

Every values within the table is the mean value of the kth attributes for ath class.

$$\mathrm{IT}_{ak}^{\mathrm{mean}} = \frac{1}{n_t^a} \sum_{j=1}^{n_t^a} d_t^{jk} \tag{6}$$

where n_t^a is the number of data samples belonging to the ath class at time interval t and d_t^{jk} is the data value corresponding to the jth data of kth attributes at time interval t. The value of f_t^{jk} is found by comparing the data value d_t^{jk} with the class information a. If the data value corresponds to the class information a, then f_t^{jk} is equal to one. Other wise, zero is assigned to it.

$$f_t^{jk} = \begin{cases} d_t^{jk}; & \text{if } d_t^{jk} \in a \\ 0; & \text{otherwise} \end{cases} \tag{7}$$

Similarly, information table belonging to the variance $\mathrm{IT}_t^{\mathrm{var}}$ is computed by finding the variance of the kth attributes for ath class.

$$\mathrm{IT}_t^{\mathrm{var}} = \left\{\mathrm{IT}_{ak}^{\mathrm{var}}; 1 \le a \le c; 1 \le k \le m_t\right\} \tag{8}$$

The formula used to compute the variance of the input data stream is given as follows:

$$\mathrm{IT}_{ak}^{\mathrm{var}} = \frac{1}{n_t} \sum_{j=1}^{n_t} \left(f_t^{jk} - \mathrm{IT}_{ak}^{\mathrm{mean}}\right)^2 \tag{9}$$

Classification Once the GNBC model is constructed, the classification can be performed by finding the posterior probability of the incoming data d_t^x with respect to every class. The class which have the maximum posterior probability is the class of the input data d_t^x. The posterior probability is computed as follows:

$$C\left(d_t^x\right) = \max_{a=1}^{C} \mathrm{posterior}\left(C_a \left| d_t^x \right.\right) \tag{10}$$

The posterior probability for the input data d_t^x with respect to the class C_a is computed based on the conditional probability of every attributes with respect to the class and evidence.

The following formula is used to compute the posterior probability and evidence.

$$\text{posterior}\left(C_a \left| d_t^x \right.\right) = \frac{P\left(C_a\right) * \Pi_{k=1}^{m_t} P\left(A_k^t \left| C_a \right.\right)}{\text{Evidence}} \tag{11}$$

where $P(C_a)$ is the probability of occurrence for the class C_a and $P(A_k^t|C_a)$ is the conditional probability of the attribute A_k^t with the class C_a. Evidence is the summation of the posterior probability of every class with respect to the input data.

$$\text{Evidence} = \sum_{a=1}^{C} \text{posterior}\left(C_a \left| d_t^x \right.\right) \tag{12}$$

The formulae used to compute the conditional probability of the attribute A_k^t with the class C_a is given as follows.

$$P\left(A_k^t \left| C_a \right.\right) = \frac{1}{\sqrt{2\pi * \text{IT}_{ak}^{\text{var}}}} * \exp\left(\frac{-\left(d_t^x - \text{IT}_{ak}^{\text{mean}}\right)^2}{2 * \text{IT}_{ak}^{\text{var}}}\right) \tag{13}$$

where $\text{IT}_{ak}^{\text{var}}$ is the variance of kth attribute of ath class. $\text{IT}_{ak}^{\text{mean}}$ is the mean of kth attribute of ath class and d_t^x is the input data to be tested.

4.2 Dynamic Updating of GNBC Model

This section presents the dynamic updating of GNBC model for the new data stream with three important steps, such as, detecting COD by rough set theory [24], updating of GNBC model and updating of important features. Once the new data stream is arrived for the classification at time interval t, the data are classified based on the updated model available at $\text{IT}_{1:t}$ and it is updated to $\text{IT}_{1:t+1}$ after knowing the class information. Here, the assumption is that at the time interval, $t+1$, the class label of the previous data stream is known.

4.2.1 Detecting concept change by rough set theory

Once a new data is arrived at time interval t, the classification is performed based on the updated model available at $\text{IT}_{1:t}$. After that, the model should be updated based on the newly arrived data. The updating of model happen only if the newly arrived data have the concept drift which means that the boundary of the data is changed heavily either inside or outside. So, detecting the change of the concept is required to decide whether the model can be updated or not. Here, the detecting concept change is performed using the lower approximation and upper approximation which is taken from the rough set theory. Here, lower approximation points out the samples lying within the boundary and upper approximation points out the samples lying outside the boundary.

The lower approximation $\underline{PY}$ is the union of all equivalence classes in $[y]_a$ which are contained by the target set, and upper approximation $\overline{PY}$ is the union of all equivalence classes in $[y]_a$ which have non-empty intersection with the target set.

$$\underline{PY} = \left\{y \left| [y]_a \leq Y \right.\right\} \tag{14}$$

$$\overline{PY} = \left\{y \left| [y]_a \cap Y \neq 0 \right.\right\} \tag{15}$$

The accuracy of approximation is the ratio of lower approximation $\underline{PY}$ and upper approximation $\overline{PY}$.

$$\text{COD}(Y) = \frac{\underline{PY}}{\overline{PY}} \tag{16}$$

Once the accuracy of approximation is estimated, it is compared with the predefined threshold called T_{COD}. If it is less than the threshold ($\text{COD}(y) < T_{\text{COD}}$), then the model should be updated.

4.2.2 Updating GNBC model

The updating of GNBC model is the way of updating the information table without utilizing the historic data. The information table available at time interval $\text{IT}_{1:t}$ is taken to update the mean and variance based on the new data d_t. This is denoted as follows,

$$\text{IT}_{1:t+1} = \left\{\text{IT}_{1:t+1}^{\text{mean}}, \text{IT}_{1:t+1}^{\text{var}}\right\} \tag{17}$$

Every values of the information table belonging to the mean is updated using the following equation which considers the variable $n_{1:t}$ which is the count of the data from the time interval 1 to t. The updated information table belonging to mean is given as follows,

$$\text{IT}_{1:t+1}^{\text{mean}} = \frac{\text{IT}_{1:t}^{\text{mean}} * n_{1:t} + \text{IT}_{t+1}^{\text{mean}} * n_{t+1}}{n_{1:t} + n_{t+1}} \tag{18}$$

where $\text{IT}_{1:t}^{\text{mean}}$ is the information table belonging to mean at time interval $t, n_{1:t}$ is the count of the data from the time interval 1 to $t, \text{IT}_{t+1}^{\text{mean}}$ is the information table constructed only based on the new data at time interval $t+1, and n_{t+1}$ is the count of the data at the time interval $t+1$. Similarly, the information table belonging to variance is updated based on the following equation by considering $n_{1:t}$ and n_{t+1}.

$$\text{IT}_{1:t+1}^{\text{var}} = \frac{\text{IT}_{1:t}^{\text{var}} * n_{1:t} + \text{IT}_{t+1}^{\text{var}} * n_{t+1}}{n_{1:t} + n_{t+1}} \tag{19}$$

where $\text{IT}_{1:t}^{\text{mean}}$ is the information table belonging to mean at time interval $t, n_{1:t}$ is the count of the data from the time interval 1 to $t, and \text{IT}_{t+1}^{\text{mean}}$ is the information table constructed only based on the new data at time interval $t+1$.

4.2.3 Updating of important features

The feature selection is an important step to be carried out to reduce the dimension of the data. In order to reduce the dimension, we have presented a feature evaluation function called entropy [6] which evaluates every feature with class attribute. Feature Evaluation Function calculates the importance of a feature given in the new database. Instead of evaluating many different subsets, it is common to select the features that have the higher degree of significance to compose the final subset of features.

$$F(a_t^k) = -\sum_{i=1}^{u(a_t)} P_i \log(P_i) \tag{20}$$

After computing the entropy for every feature value, the top features having the less value in entropy are taken as for the further steps. Where a_t^k is attribute vector and $u(a_i)$ is number of unique values in attribute vector.

4.3 Classification Using Updated GNBC Model Considering Recurring Concept Drift

The final step is to perform the classification using the updated GNBC model by considering recurring concept drift. The newly data arrived at time interval $t+1$ can be denoted as d_{t+1} which contain the multiple attributes. The attributes can be a new attributes or the old attributes because we considered here the recurring dimensional space. If a new attribute is added, the mean and variance of the attributes is additionally added into the information table. If the data attributes belonging to the old sample, the mean and variance belonging to those attributes are only taken from the information table for performing the classification. The information table updated based on new data samples can be represented as follows,

$$\mathrm{IT}_{1:t+1} \Rightarrow \mathrm{IT}_{1:t+1}^{\mathrm{Rec}} \tag{21}$$

The reduced information table based on the recurring dimensional space is used to predict the class label of the new data using posterior probability of the objective function.

$$C\left(d_{t+1}^x\right) = \underset{a=1}{\overset{C}{\mathrm{Max}}}\ \mathrm{posterior}\left(C_a \left| d_{t+1}^x \right.\right) * O_t \tag{22}$$

The objective function O_t considers, sensitivity, specificity and accuracy values as like follows,

$$O_t = \frac{1}{3}(E_t + \mathrm{Sen}_t + \mathrm{Spec}_t) \tag{23}$$

where E_t is the accuracy related to time interval t, Sen_t is sensitivity, and Spec_t is the specificity related to time interval t. The value of E_t is computed as follows,

$$E_t = \frac{1}{t}\sum_{t=1}^{t} \mathrm{PE}_t \tag{24}$$

$$\mathrm{PE}_t = \frac{n_t^c}{n_t} \tag{25}$$

where n_t^c is the number of data samples correctly classified at time interval t and n_t is the number of data samples at time interval t. Figure 2 shows the Algorithmic description of the proposed RGNBC model.

5 Results and Discussion

This section presents the experimental results of the proposed RGNC model and the comparative analysis with the existing works using sensitivity, specificity and accuracy.

5.1 Dataset Description

The experimentation is performed using two large databases such as, Skin Segmentation Data Set and Localization Data which are taken from UC Irvine Machine Learning [25]. *Skin Segmentation Data Set (database 1):* The skin dataset is collected by randomly sampling the B, G, R values from face images of various age groups (young, middle, and old), race groups (white, black, and asian), and genders obtained from FERET database and PAL database. The total number of instances in this data is 245,057; out of which 50,859 is the skin samples and 194,198 are non-skin samples. *Localization Data for Person Activity Data Set (database 2):* This database is collected from the people who used for recording of the data by wearing four tags (ankle left, ankle right, belt and chest). Each instance is a localization data for one of the tags. The tag can be identified by one of the attributes. The total number of instances is 164,860.

5.2 Evaluation Metrics

The performance is evaluated using sensitivity, specificity and accuracy metrics. Sensitivity is the proportion of true positives that are correctly identified by a diagnostic test. It shows how good the test is at detecting a disease. Specificity is the proportion of the true negatives correctly identified by a diagnostic test. It suggests how good the test is at identifying normal (negative) condition. Accuracy is the proportion of true results, either true positive or true negative, in a population. It measures the degree of veracity of a diagnostic test on a condition. These measures can be expressed in the terms of TP, FP, FN and TN as like the Eqs. (26), (27) and (28).

Fig. 2 Algorithmic description of the proposed RGNBC model

```
3        T_COD →CoD threshold
4   Output:
5        C →Classified label
6   Procedure
7   Begin
8        Divide database D into a chunk of data, d_t
9        For t = 1 : N
10           Read d_t
11           If(t==1)
12               Build GNBC model IT_t
13           Endif
14           Perform classification of d_t using IT_1:t
15           If(t!=1)
16               Find accuracy of approximation COD(Y)
17               If (COD(Y) < T_COD)
18                   Update GNBC model IT_1:t+1
19               Endif
20           Endif
21       Endfor
22           Return Classified label C
23       Endfor
24  End
```

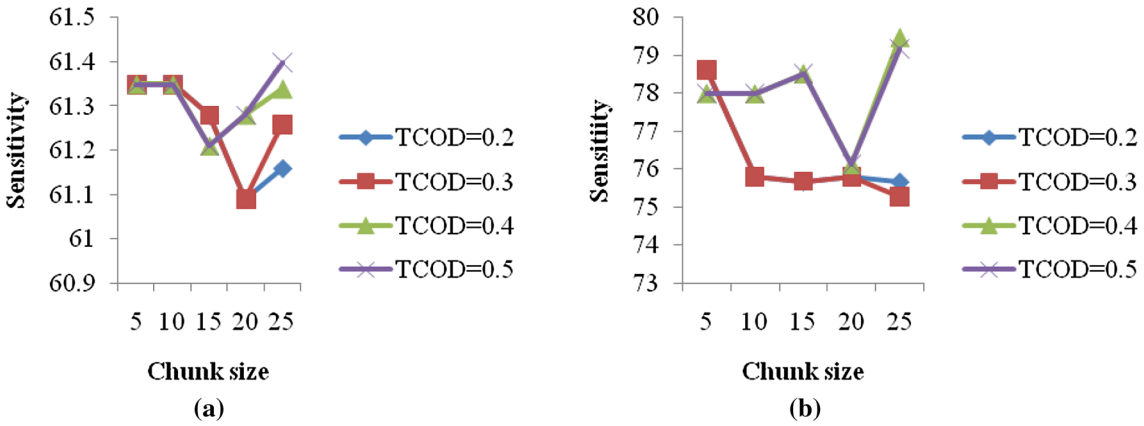

Fig. 3 Sensitivity graph. **a** Skin data and **b** localization data

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \tag{26}$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{27}$$

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP}) \tag{28}$$

where true positive (TP) is correctly identified, false positive (FP) is incorrectly identified, true negative (TN) is correctly rejected, and false negative (FN) is incorrectly rejected.

5.3 Experimental Set Up

The proposed RGNBC model is implemented using using Java 1.7 with NetBeans IDE 7.3. The system has i5 processor of 2.2 GHz CPU clock speed with 4 GB RAM and 64 bit operating system running with Windows 8.1. The performance of the proposed method is evaluated using sensitivity, specificity and accuracy with the existing system, called MReC-DFS [5]. The important parameter of COD (T_{COD}) is extensively analyzed to find the best value for the comparative analysis.

5.4 Performance Evaluation of the Proposed Algorithm

This section presents the performance evaluation of the proposed RGNBC model. Figure 3a shows the sensitivity graph for the skin data. Here, the performance is analyzed for the four different of COD threshold (T_{COD}) by fixing from 0.2 to

0.5. The results are analyzed by varying the chunk size. For the less number of chunk size, the performance is high and then, performance is decreased. Finally, the performance is again increased when the chunk size is increased. For the skin data, the maximum sensitivity of 61.4 % is achieved when the number of chunk is fixed to 25 with the COD threshold of 0.5. Figure 3b shows the sensitivity graph for the localization dataset. The similar kind of performance as like the skin data is obtained in this figure also. Initially, the sensitivity is high and then decreased to specific value. Finally, the values are again high. The highest sensitivity obtained by the proposed RGNBC model is 79.47 % for the chunk size of 25 with the COD threshold of 0.4.

Figure 4a shows the specificity graph for the skin data. Here, the performance is analyzed for the various chunk sizes of 5, 10, 15, 20 and 25. When the chunk size is increased, the

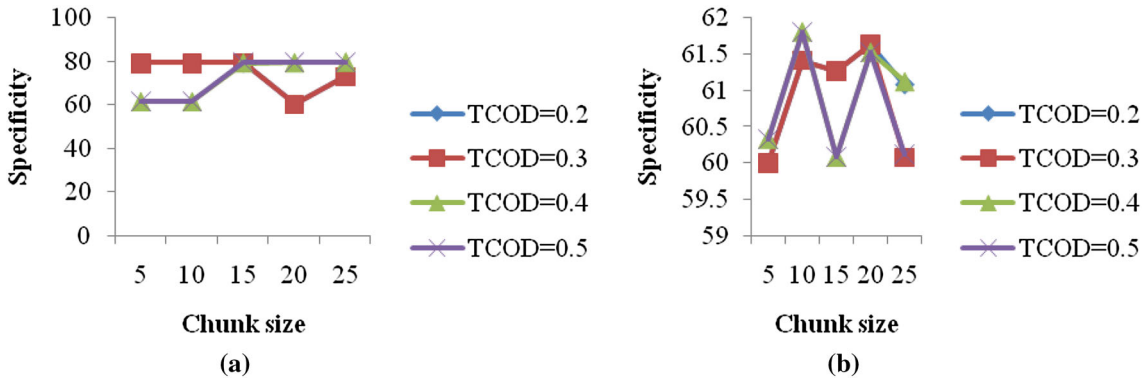

Fig. 4 Specificity graph. **a** Skin data and **b** localization data

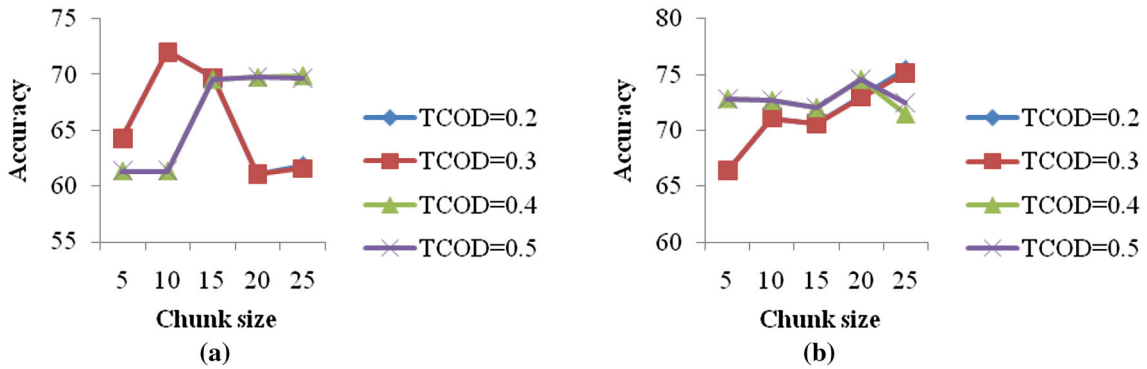

Fig. 5 Accuracy graph. **a** Skin data and **b** localization data

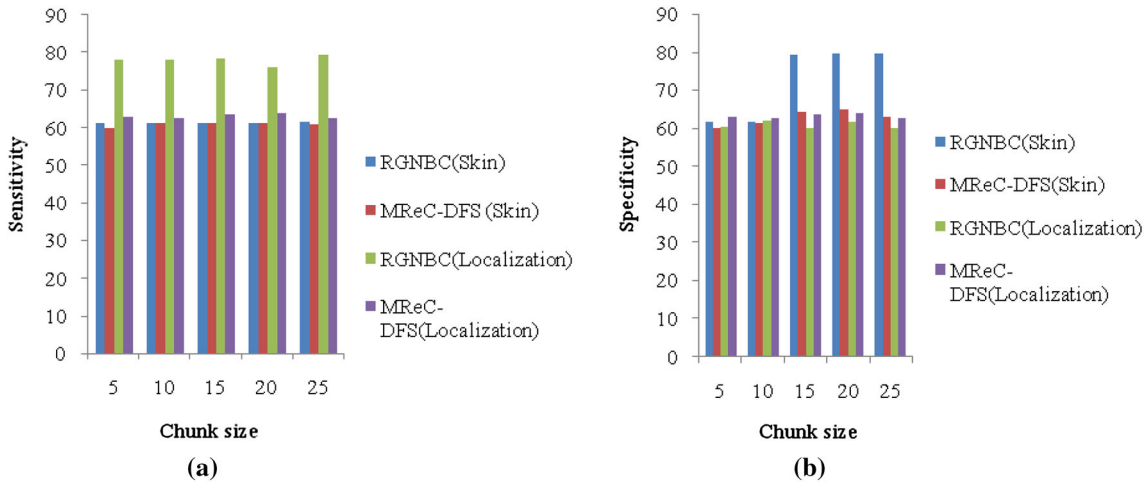

Fig. 6 Comparative graph. **a** Sensitivity and **b** specificity

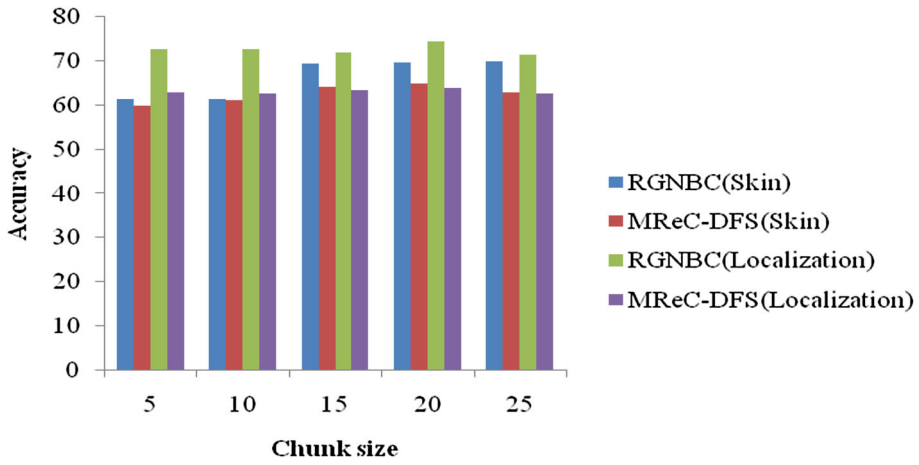

Fig. 7 Accuracy graph for comparative analysis

performance is decreased for the threshold of 0.2 and 0.3. For the COD threshold of 0.2, the maximum accuracy is 79.43 % and the accuracy value of 79.71 % is reached when the COD threshold is equal to 0.4. Similarly, specificity graph for the localization data is plotted in Fig. 4b. From the figure, we understand that the performance is much fluctuated over the sizes of chunk. When the COD threshold is fixed to lower value of 0.2, the specificity of 60 % is obtained for the chunk size of 5 but we obtained 61 % for the chunk size of 25. The maximum specificity of 61.8 % is obtained for the COD threshold of 0.4.

Figure 5a shows the accuracy graph for the skin data. For the COD threshold of 0.2, the accuracy of 64.29, 72.08, 69.7, 61 and 61.8 % is obtained for the chunk size of 5–25. Similarly, for the COD threshold of 0.5, the accuracy of 61.35, 61.35, 69.59, 69.7 and 69.63 % is obtained for the chunk size of 5–25. The accuracy graph is low at the initial state, and the values are increased for the increasing number of chunk size. After that, the accuracy values are again decreased until the chunk size is 20 for the COD threshold of 0.3. The maximum accuracy in the skin data attained is 72.08 % for the chunk size of 10. Figure 5b shows the accuracy graph for the localization data. For the COD threshold of 0.2, the accuracy of 66.36, 71.11, 70.59, 72.99 and 75.44 % is obtained for the chunk size of 5–25. The maximum accuracy attained by the proposed RGNBC in localization data is 74.56 %.

5.5 Comparative Analysis

This section shows the comparative analysis of the proposed RGNBC model with the existing MReC-DFS [5]. Figure 6a shows the comparative of these two techniques in both the datasets using sensitivity measure. From the figure, we prove that the proposed RGNBC model obtained high sensitivity in both the datasets than the MReC-DFS. For the skin data, the maximum sensitivity obtained by the proposed RGNBC model is 61.4 % which is higher than the existing MReC-DFS which obtained only 61 %. Similarly, in localization data, the proposed RGNBC obtained the maximum sensitivity of 79.47 % than the 63.97 % which is obtained by the existing method. Figure 6b shows the comparative graph of specificity for both the datasets. Here, the accuracy of 61.53, 61.53, 79.34, 79.43 and 79.71 % is obtained for the chunk size of 5–25. The maximum value of 79.71 % is reached by the proposed RGNBC model when the chunk size is equal to 25. Figure 7 shows the accuracy graph of the proposed RGNBC model and MReC-DFS for the various chunk sizes. The proposed RGNBC model attained the maximum accuracy of 69.9 and 74.5 % for the skin data and localization data where the existing obtained the maximum accuracy of 64.9 and 63.9 %. This graph ensured that the proposed RGNBC model outperformed the existing algorithm in two datasets even for the various sizes of chunks.

6 Conclusion

We have developed a new dynamic model for handling the recurring concept drift in data stream classification. Here, RGNBC is newly developed by integrating the rough set theory with naïve bayes classifier. Here, the process of detecting the concept drift and the updating of classification model is newly performed. For the detection of concept drift, accuracy of approximation based on rough set theory was utilized for detecting the concept drift and the naïve bayes classifier is dynamically updated based on the new mathematical model to handle the recurring concept drift. Also, the classification is performed using the posterior probability with the objective function which considers the multiple criteria. For the experimentation, the proposed RGNBC model is validated with two large datasets and the results are compared against the existing MReC-DFS algorithm using sensitivity, specificity and accuracy. The outcome ensured that the proposed RGNBC model attained the maximum accuracy of 74.5 %

while compared with the existing algorithm. In future, naïve bayes model can be replaced with a new learning mechanism to handle the recurring concept drift.

References

1. Mena-Torres, D.; Aguilar-Ruiz, J.S.: A similarity-based approach for data stream classification. Expert Syst. Appl. **41**, 4224–4234 (2014)
2. Zhang, P.; Zhou, C.; Wang, P.; Gao, B.J.; Zhu, X.; Guo, L.: E-Tree: an efficient indexing structure for ensemble models on data streams. IEEE Trans. Knowl. Data Eng. **27**(2), 461–474 (2015)
3. Rutkowski, L.; Jaworski, M.; Pietruczuk, L.; Duda, P.: Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans. Knowl. Data Eng. **26**(1), 108–119 (2014)
4. Brzezinski, D.; Stefanowski, J.: Reacting to different types of concept drift: the accuracy updated ensemble algorithm. IEEE Trans. Neural Netw. Learn. Syst. **25**(1), 81–94 (2014)
5. Gomes, J.B.; Gaber, M.M.; Sousa, P.A.C.; Menasalvas, E.: Mining recurring concepts in a dynamic feature space. IEEE Trans. Neural Netw. Learn. Syst. **25**(1), 95–110 (2014)
6. Pinheiro, R.H.W.; Cavalcanti, G.D.C.; Ren, T.I.: Data-driven global-ranking local feature selection methods for text 4 categorization. Expert Syst. Appl. **42**(4), 1941–1949 (2015)
7. Rish, I.: An empirical study of the naive Bayes classifier. In: Proceedings of IJCAI Workshop on Empirical Methods in AI (2001)
8. Alippi, C.; Liu, D.; Zhao, D.; Bu, L.: Detecting and reacting to changes in sensing units: the active classifier case. IEEE Trans. Syst. Man Cybern. Syst. **44**(3), 353–362 (2013)
9. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: Proceedings of ACM SIGKDD 10th International Conference Knowledge Discovery and Data Mining, pp. 128–137 (2004)
10. Gao, J.; Fan, W.; Han, J.: On appropriate assumptions to mine data streams. In: Proceedings of IEEE Seventh Internationall Conference on Data Mining (ICDM), pp. 143–152 (2007)
11. Hulten, G.; Spencer, L.; Domingos, P.: Mining time-changing data streams. In: Proceedings of ACM SIGKDD Seventh Internationall Conference on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
12. Kolter, J.; Maloof, M.: Using additive expert ensembles to cope with concept drift. In: Proceedings of 22nd International Conference on Machine Learning (ICML), pp. 449–456 (2005)
13. Wang, H.; Fan, W.; Yu, P.S.; Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of ACM SIGKDD Ninth Int'l Conference on Knowledge Discovery and Data Mining, pp. 226–235 (2003)
14. Katakis, I.; Tsoumakas, G.; Vlahavas, I.: On the utility of incremental feature selection for the classification of textual data streams. In: Advances in Informatics. Springer, New York, NY, pp. 338–348 (2005)
15. Gomes, J.B.; Menasalvas, E.; Sousa, P.: Tracking recurrent concepts using context. In: Proceedings of 7th International Conference on RSCTC, pp. 168–177 (2010)
16. Gama, J.; Kosina, P.: Tracking recurring concepts with metalearners. In: Proceedings of 14th Portuguese Conference on Artificial Intelligence, p. 423 (2009)
17. Yang, Y.; Wu, X.; Zhu, X.: Mining in anticipation for concept change: proactive–reactive prediction in data streams. Data Min. Knowl. Discov. **13**(3), 261–289 (2006)
18. Al-Khateeb, T.; Masud, M.M.; Khan, L.; Aggarwal, C.; Hans, J.; Thuraisingham, B.: Stream classification with recurring and novel class detection using class-based ensemble. In: Proceedings of IEEE 12th International Conference on Data Mining (ICDM), pp. 31–40 (2012)
19. Masud, M.M.; Al-Khateeb, T.M.; Khan, L.; Aggarwal, C.; Gao, J.; Han, J.; Thuraisingham, B.: Detecting recurring and novel classes in concept-drifting data streams. In: Proceedings of IEEE 11th International Conference on Data Mining (ICDM), pp. 1176–1181 (2011)
20. Gama, J.; Kosina, P.: Recurrent concepts in data streams classification. Knowl. Inf. Syst. **40**(3), 489–507 (2014)
21. Sripirakas, S.; Pears, R.: Mining recurrent concepts in data streams using the discrete Fourier transform. Proc. Data Warehous. Knowl. Discov. **8646**, 439–451 (2014)
22. Masud, M.M.; Chen, Q.; Khan, L.; Aggarwal, C.C.; Gao, J.; Han, J.; Srivastava, A.; Oza, N.C.: Classification and adaptive novel class detection of feature-evolving data streams. IEEE Trans. Knowl. Data Eng. **25**(7), 1484–1497 (2013)
23. Abdulsalam, H.; Skillicorn, D.B.; Martin, P.: Classification using streaming random forests. IEEE Trans. Knowl. Data Eng. **23**(1), 22–36 (2011)
24. Pawlak, Z.: Rough sets. Int. J. Parallel Prog. **11**(5), 341–356 (1982)
25. UC Irvine Machine Learning Repository from http://archive.ics.uci.edu/ml/datasets.html