HRFuzzy: Holoentropy-Enabled Rough Fuzzy Classifier for the Classification of Evolving Data Streams

Jagannath Nalavade, Senthil Murugan
Veltech Dr. RR & Dr. SR Technical University
India
{jeganen@gmail.com} {senthilmuruganme@gmail.com}

ABSTRACT: *Due to continuous growth of recent applications such as, telecommunication, sensor data, financial applications, analyzing of data streams, conceptually endless sequences of data records, frequently arriving at high rates is important task among the data mining community. Among the various task involved in data stream analysis, the classification of data streams pose various challenging issues compared to popular data classification algorithms. Since the classification algorithm performs endlessly, it must be able to adapt the classification model to handle the change of concept or boundaries between classes. In order to handle these issues, we have developed a new fuzzy system called, HRFuzzy to classification of evolving data streams. Here, rough set theory and holoentropy function are utilized to construct the dynamic classification model. In the fuzzy system, the rules are generated using k-means clustering and membership functions are dynamically updated using holoentropy function. The experimentation of the proposed HRFuzzy is performed using two different databases such as, skin segmentation dataset and localization data and the performance is compared with adaptive k-NN classifier in terms of accuracy and time. From the outcome, we proved that the proposed HRFuzzy outperformed in both the metrics by giving the maximum performance.*

Keywords: Data Stream, Classification, Fuzzy, Rough Set, Holoentropy, Concept Change

Received: 19 October 2019, Revised 28 December 2019, Accepted 8 January 2020

DOI: 10.6025/jes/2020/10/1/34-47

1. Introduction

In machine learning, pattern recognition and data mining, learning classifiers from the data is a major task. The classical methods works better in static environments, in which the entire dataset is presented to the learning algorithm. The dataset is stored in electronic format and the algorithm can use it whenever required. The target concepts which require learning are also predetermined. For static classification, several solutions and classifiers are available recently. But some recent applications such as sensor networks, traffic management and telecommunication, [5] the learning algorithms performs well in dynamic environments where the data is not stored and it is generated continuously.

For such applications, data classification becomes a main task in data stream mining field. The size of data stream [20-23] is infinite. The data elements continuously enter the system with high rate. Furthermore, the data concept can develop with time, called as concept drift [3, 4]. Due to concept drift, storage of data streams in main memory is not possible. In such environment, storage, querying and mining become a difficult task. This feature is associated with the computational resources to evaluate large volume of data and it is studied in detail in the literature where many proposals are introduced to present better algorithms. One of the proposals is to process the data streams in online which make sure that the results are updated and the queries can be replied without delay [1].

As the data stream occur continuously and fast, many incremental learners are intended to deal with this problem [11, 9]. Additionally, when the concepts of stream alter with time, concept-drift occurs. In literature, several techniques are available to deal with the problem of concept-drift in data stream classification [10], [12, [13]. For example, many techniques make an assumption that the basic feature space is static one, which is not possible in real world applications [16] where the features and the significance to the target concept may vary. Also, as the concepts reappear, the accuracy and the processing time of the learning process is improved by reusing the models which was already learned [14], [15], [17].

In this paper, multi-classification of data streams is done by automatically detecting the concept drift. Here, rough and fuzzy set theory are combined for data stream multi-classification. The classification is performed using fuzzy rule classifier where, the membership function designing and the rule definition are two important steps. For a dynamic data stream, these two processes should be updated dynamically based on the characteristics of new data. The updating behavior of membership function is done using rough set theory which dynamically extends the boundary regions based on the arrival of new data and the membership function is updated based on the interval. In the second step, the rules are continuously updated using the proposed holoentropy-based method. Based on concept drift and feature space, the updating of membership function and rule definition is done using rough set and holoentropy.

The main contribution of the paper is given as follows,

• A new fuzzy system called, HRFuzzy (Holoentropy-enabled Rough Fuzzy classifier) is developed by combining the holoentropy [24], rough set [18] and fuzzy set theory [25].

• We make use of rough set theory for change of detection process.

• The dynamic updating of fuzzy membership function is done by detecting the concept change and the formulae devised in this paper.

• The fuzzy rules are changed continuously using k-means clustering [26] and the weight of the rules in which the formulae is devised newly.

The paper is organized as follows. Section 2 presents the literature review and section 3 presents the motivation behind the approach. Section 4 presents holoentropy-enabled rough fuzzy classifier to classification of evolving data streams and section 5 discusses the results and discussion. Finally, conclusion is given in section 6.

2. Literature Review

Table 1 discusses the review of different data stream classification methods available in the literature. Most of methods are modifying the traditional classifier to adapt to concept drift as like, adaptive k-nearest neighbor [2], McDiarmid tree algorithm [2]. Some authors bring different strategy for detecting the concept drift and then, the classification is done with the popular classification algorithm. In table 1, we discuss the recent data stream classification methods and their advantages with the major issues.

Authors	**Contribution**	**Advantages**	**Disadvantages**
Dayrelis Mena-Torres and Jesús S. Aguilar-Ruiz [1]	Similarity-based approach	Advantages of the Instance-based Learningtechniques	Refine themodel to detect and deal with only abrupt concept changes

Authors	Contribution	Advantages	Disadvantages
Cesare Alippi *et al.* [2]	Adaptive k-nearest neighbor and support vector machine classifiers	Exactly detecting change trend	It is more suitable for numerical data
Peng Zhang *et al.* [3]	Tree-based indexing structure	Reinsertion is possible incrementally	Maintenance of tree model and its storage space requirement is high
Leszek Rutkowski *et al.* [4]	McDiarmid tree algorithm	It is easy to construct and handle the concept drift	Difficult in handling abnormally distributed data
Dariusz Brzezinski and Jerzy Stefanowski [5]	Accuracy-based weighting mechanisms	Consider the periodic weighting mechanism	Adapting weight for different data space seems tough
João Bártolo Gomes *et al.*[6]	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
Mohammad M. Masud *et al.* [7]	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 fromthe actual arrival of a novel class.
Hanady Abdulsalam *et al.* [8]	Combines the ideas ofstreaming decision trees and Random Forests	It quicklyrecords the new expected classification accuracy after the changes are presented in the stream	Handling multiple classes with this hybrid model is difficult

Table 1. Literature review

3. Motivation Behind the Approach

This section discusses the problem formulated for data stream classification and the major challenges to be addressed to design a good data stream classification method.

3.1 Problem Definition

Let assume that the input database, D is partitioned into a chunk of data samples d_t having a size of w, $D = \{d_t; 0 \leq t \leq w\}$. At the current time, the chunk data d_t is read out and classification is to be performed on the data. Every data objects in the chunk is represented with n attribute vector a_j, $d_t \in a_j; 1 \leq j \leq n$. The challenge here is that the class of the data to be identified for this current chunk data. When the data classification is performed on the current chunk data d_t, the class labels of the previous chunk, d_{t-1} are known but the data of previous samples other than d_{t-1} cannot be stored or preserved only the classification model has to modified or updated accordingly.

3.2 Challenges

Due to evolving nature of data streams, learning of classifier cannot be performed with full data space but standard data mining algorithm requires scanning of databases every time. But, for data stream, multiple scanning of database of building learning model is not possible practically. So, the extraction of model from the evolving data stream without multiple scanning within the original data space is an important challenge to be considered.

Due to dynamic nature of data stream, not only the size of the records is increasing but also the dimension of the data space or 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.

Also, the newly arriving data cannot be in original boundary defined for every feature so classifier model should consider this drift in boundary when developing data stream classification.

To maximize usefulness, classification should be possible as soon as a sufficiently robust model has been built. The model should be updated because of a change in the underlying data. Hence, the classification algorithm should be incremental, so that changes require a model update rather than a completely new model.

Most algorithms handle data with only numerical or categorical attributes but the real datasets contains both numerical and categorical data. It is, therefore, essential to consider data with both attributes.

In [2], Change detection tests (CDTs) was designed to inspect structural changes in industrial and environmental data and adaptive k-nearest neighbor and suitably retrained when the change is detected. This method removes the old data frequently so the preservation of historical information in the classification process is limited.

4. Proposed Methodology: Holoentropy-enabled Rough Fuzzy Classifier to Classification of Evolving Data Streams

This section presents the HRF system to classification of evolving data streams. Figure 1 shows the block diagram of the proposed holoentropy enabled rough fuzzy system. The block diagram contains the two important processes such as k-means enabled fuzzy system and updating of fuzzy system. In the k-means enabled fuzzy system, k-means clustering is used for extracting the rules and the updating of system is done using the holoentropy function. In updating fuzzy membership function, the change of detection test is done using rough set theory. The membership function and rules are updated if the COD test is detected the concept drift. The membership function is then updated using rough set approximations and the fuzzy rules are updated using the holoentropy function.

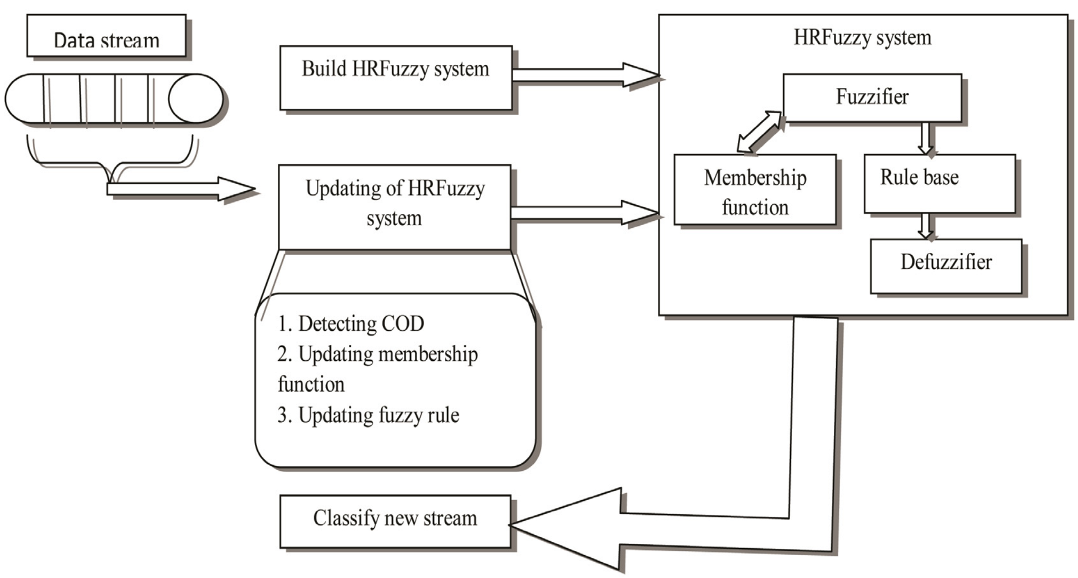


Figure 1. Block diagram of the HRFuzzy system

4.1 Building K-means enabled Fuzzy System

In this work, missing values are replaced with average value of the attribute and categorical values are represented with the numerical values by replacing every unique value with unique numerical value. Then, classifier model is built based on the first data stream d_0 and continuously updating the classifier model based on the previous data stream d_{t-1} because we know the class label for this data stream only in the current time. Accordingly, fuzzy classifier is to be constructed by devising the membership function and the fuzzy rules.

4.1.1. Fuzzy Membership Function

The initial step in the fuzzy classifier is to devise the membership function which is used to convert the data values into linguistic variables based on the degree of membership. The designing of membership function is a way to construct the curve based on the data behavior. In this paper, we utilized triangular membership function to define the data characteristics and every attributes is defined with three membership function. Accordingly, the membership function can be indicated as, $\mu_{jk}^{(0)}$ where, *j* refers to the index of the attribute, *k* refers to the index of membership function. The membership function of the j^{th} attributes of i^{th} membership function is defined as follows,

$$\mu_{jk}^{(0)} = f\left(d_0^{\,ij}; b, c, g\right) \tag{1}$$

The definition of the triangular membership functions is given as follows:

$$f\left(d_0^{\,ij}; b, c, g\right) = \left\{ \begin{array}{ll} 0; & d_0^{\,ij} \le b \\ \dfrac{d_0^{\,ij} - b}{c - b}; & b \le d_0^{\,ij} \ge c \\ \dfrac{g - d_0^{\,ij}}{g - c}; & c \le d_0^{\,ij} \ge g \\ g; & g \le d_0^{\,ij} \end{array} \right\} \tag{2}$$

In order to design the triangular function based on the above equation, the values of *b, c* and *g* can be computed by taking the minimum, centre and maximum value from the attribute vector.

$$b = d_i^{\min} \tag{3}$$

$$c = \frac{d_i^{\min} + d_i^{\max}}{2} \tag{4}$$

$$g = d_i^{\max} \tag{5}$$

Where, $d_i^{\min}$ is the minimum value of the i^{th} attribute and $d_i^{\max}$ is the maximum value of the i^{th} attribute. This process is repeated for all the attributes and their corresponding membership function is devised automatically. Figure 2 shows the sample triangular membership function.

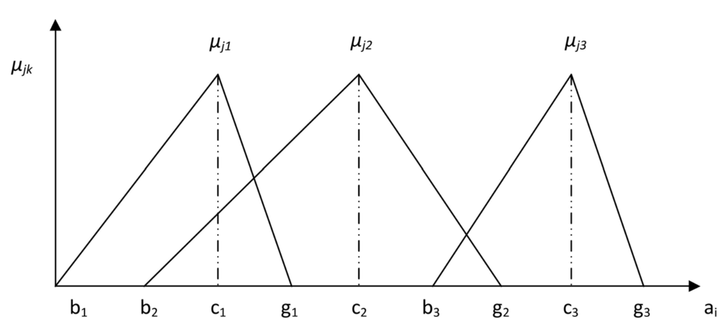


Figure 2. Triangular membership function

4.1.2. Fuzzy Rule base

Fuzzy rule base is the second step for fuzzy-based classification system. The generation of fuzzy rules is very challenging. Here, we utilize the traditional clustering algorithm, called k-means clustering for generation fuzzy rules. The steps involved in this step are given as follows:

1) Discretization of the Input Data: The input chunk data d_0 is discretized into three set of linguistic values as we have utilized three membership functions.

2) Data Partitioning: Based on the class information, the chunk data is partitioned into a set of partitions where, the number of partitions is equal to the number of classes.

3) Applying K-means Clustering: Based on the number of rules required per every class, the k-means clustering is applied for every partition and *k*-clusters is found from the every partition.

Let assume that $d_0{}^i$ be *i* the data object of the first chunk data after discretization. The objective is to find out the k centroids which are then assumed to be the rules. At first, centroids are randomly taken from the data. The centroids are represented as, $\{C_1, C_2, ..., C_k\}$.

Once centroids are randomly found out, the distance between every data object with the centroids are computed and the data objects are grouped based on the minimum distance contributed by the centroid. Then, for the new cluster, centroid is computed based on the following equation. The above process is repeated until there is no movement in the clusters.

$$C_i = \frac{1}{N_{ci}} \sum_{j=1}^{N_{ci}} d_0{}^i \tag{6}$$

4) Obtaining Fuzzy Rules: The data which is nearer to k- centroid is taken as the rule which is then converted to fuzzy format. Suppose, the clustering is applied on the first class partitions and the centroid looks like, [1 2 1 1] then fuzzy rules can be defined as IF A1 is LOW OR A2 is HIGH OR A3 is LOW OR A4 is LOW THEN OUTPUT is HIGH.

$$C_i \in d_0{}^i \Rightarrow \{R_{ci}\} \tag{7}$$

Where, R_{ci} is fuzzy rules obtained from c^{th} class. Finally, the rules belonging to all the classes are combined to constitute the final rule set, R_0.

$$\{R_{ci}\} \Rightarrow R_0 \tag{8}$$

Rule Weighting: Let us assume that every rules presented in the set R_0 is indicated as, $R^0 = \{R_1^0 \ R_2^0 \ R_{c*k}^0\}$. The weight of every rules is computed by finding the matching count of the rules within the discretized database. The formulae to compute the weight of the rule, R_i^t is given as follows:

$$w(R_i^0) = \left(\frac{\sum_{i=1}^{c*k} M\left(R_i^0, d_0{}^{dis}\right)}{c*k*n} \right) \tag{9}$$

Where, $d_0{}^{dis}$ is the discretized database of d_0 and $M\left(R_i^t, d_0{}^{dis}\right)$ is the matching count of the rule within the discretized database.

4.1.3. Data Classification using Fuzzy System

Once the membership function and rule base are defined for the fuzzy system, the classification can be done by inputting the stream of data. The devised fuzzy system can be represented as $F_{d_0}\{\mu_{jk}^{(i)}, R_0\}$. The fuzzy system generates a score value which can be then used for classifying the data labels.

4.2. Dynamic updating of HRFuzzy System

Once the fuzzy system is defined initially, the system is ready to do the data classification for the incoming data streams d_t. But, the important scenario is that simultaneously update the fuzzy system for the previous data stream d_{t-1} if there is any concept change because that data stream have the class label in the current time. The process of updating the HRFuzzy system for previous data stream d_{t-1} is explained here in three important steps like, detecting concept by rough set theory, updating membership function, updating fuzzy rules.

4.2.1. Detecting Concept Change by Rough Set Theory

The concept change for the previous data stream d_{t-1} is computed by rough set theory [18] which is a formal approximation of a crisp set in terms of a pair of sets which give the lower and the upper approximation of the original set. The lower approximation of a set X with respect to a is the set of all objects, which can be for certain classified as X with respect to a. The upper approximation of a set X with respect to a is the set of all objects which can be possibly classified as X with respect to a. The boundary region of a set X with respect to g is the set of all objects, which can be classified neither as X nor as not-X with respect to a. The upper and lower approximations can be defined as follows:

$$\underline{a}X = \{x | [x]_a \leq X\} \tag{10}$$

$$\bar{a}X = \{x | [x]_a \cap X \neq \phi\} \tag{11}$$

Base on the lower and upper approximation of the data, the accuracy of approximation is computed based on the following equation.

$$\alpha_a(X) = \frac{|\underline{a}X|}{|\bar{a}X|} \tag{12}$$

The accuracy of approximation is then compared with a threshold, called T. If the $\alpha_a(X)$ is less than the threshold T, there is a concept drift so updating is required.

4.2.2. Updating Membership Function

The dynamic updating of membership function is about to update the variables of b, c and g based on the drift in the previous data stream, d_{t-1}. In order to update these variables, the following two constraints should be satisfied. i) Accuracy of the approximation should be less than the threshold which means that the concept drift presented in the incoming data stream, ii) the variable values of b_{t-1}, c_{t-1} and g_{t-1} for the previous data stream d_{t-1} should be different. Accordingly, the membership functions of the j^{th} attributes of i^{th} membership function is defined as follows for the previous data stream d_{t-1}.

$$\mu_{jk}^{t-1} = f(d_{i-1}, b_{t-1}, c_{t-1}, g_{t-1}) \tag{13}$$

By making triangular membership function for the above fuzzy system, the values of b_{t-1}, c_{t-1} and g_{t-1} can be computed by comparing the previous and current variables of membership variables. The values of those variables are computed as follows. From this equation, we understand that if the concept drift is presented in the database for the recent data streams, it automatically extend or shrink the range of values in the membership function.

$$b_{t-1} = \begin{cases} b_{t-2} & ; \quad b_{t-2} = b_{t-1} \\ \min\{d_{t-2}^{\min}, d_{t-1}^{\min}\} & ; \quad else \end{cases} \tag{14}$$

$$g_{t-1} = \begin{cases} g_t & ; g_{t-2} = g_{t-1} \\ \max\{d_{t-2}^{\max}, d_{t-1}^{\max}\} & ; else \end{cases} \tag{15}$$

$$C_{t-1} = \begin{cases} C_{t-2} & ; \quad C_{t-2} = C_{t-1} \\ \dfrac{b_{t-1} + g_{t-1}}{2} & ; \ else \end{cases} \tag{16}$$

4.2.3. Updating Fuzzy Rules by Holoentropy Function

Once the membership function is updated, the next step is to update the fuzzy rules for the previous data stream based on the class label. So, the data stream d_{t-1} is directly given to the k-means clustering to mine the fuzzy rules. The fuzzy rules of the previous data stream is represented as, R^{t-1}. The updating of rules is purely based on weight of the rules and holoentropy. Here, holoentropy decide which all are the rules to be replaced and weight of the rules decides which rules to be selected to replace with. Once the rules of previous data streams R^{t-2} and R^{t-1} are found out, the holoentropy is found out among all the combination of rules from two previous data streams.

$$IHE\,(R_j^{t-2}, R_j^{t-1}) = HE\,(R_j^{t-1}) - CHE\,(R_j^{t-1}, R_j^{t-2}) \tag{17}$$

Where, $IHE\left(R_j^{t-2}, R_j^{t-1}\right)$ is informative holoentropy of rules R_j^{t-2} and R_j^{t-1}. $HE\left(R_j^{t-1}\right)$ is holoentropy of rule R_j^{t-1}. $CHE\left(R_j^{t-1}, R_j^{t-2}\right)$ is conditional holoentropy of rules R_j^{t-2} and R_j^{t-1}.

$$HE\,(R_j^{t-1}) = W \times E\,(R_j^{t-1}) \tag{18}$$

$$W = 2\left(1 - \frac{1}{1 + \exp(-E\,(R_j^{t-1}))}\right) \tag{19}$$

$$E(R_j^{t-1}) = -\sum_{i=1}^{u(R_j^{t-1})} P_i \times \log P_i \tag{20}$$

Where, R_j^{t-1} is rule vector and $u(R_j^{t-1})$ is number of unique values in rule vector R_j^{t-1}.

$$CHE\,(R_j^{t-2}, R_j^{t-1}) = \sum_{i=0}^{u(R_j^{t-1})} P_i \times HE\,(R_j^{t-2}, R_j^{t-1}) \tag{21}$$

$$HE(R_j^{t-2}, R_j^{t-1}) = W_h * E(R_j^{t-2}, R_j^{t-1}) \tag{22}$$

$$W_h = 2\left(1 - \frac{1}{1 + \exp\,(-E\,(R_j^{t-2}, R_j^{t-1}))}\right) \tag{23}$$

$$E\,(R_j^{t-2}, R_j^{t-1}) = \sum_{i=1}^{u(R_j^{t-2})} P\left(R_j^{t-2} = i, R_j^{t-1} = i\right) \times \log\left(R_j^{t-2} = i, R_j^{t-1} = j\right) \tag{24}$$

Then, the couple of rules having minimum holoentropy are selected. It means that these couple of rules is similar kind of rules in the two previous data streams. Once the most relevant rules of both data stream are identified, preserving of which rule out of these two rules is identified based on the weight of the rules.

$$R_j^{t-1} = \begin{cases} R_j^{t-2}\ ; & w\left(R_j^{t-1}\right) < w\left(R_j^{t-2}\right) \\ R_j^{t-1};\ else \end{cases} \tag{25}$$

Based on the updated membership function and rules, the fuzzy system is dynamically changed by adapting the concept of drift in the incoming data stream. The uploaded $F\left\{\mu_{jk}^{t-1}, R_{ci}^{t-1}\right\}$ is then used for data stream classification. This process of updating is simultaneously happening for every new data stream. Figure 3 derails the algorithmic description of the proposed HRFuzzy system.

1	**Algorithm:** HRFuzzy
2	**Input:** $d_t \rightarrow$ Input database
3	$k \rightarrow$ Number of rules per class

$T \rightarrow$ COD threshold

Output:

$F_{d\,t-1}(\mu_{t-1}, R_{t-1}) \rightarrow$ Updated membership function

Procedure

Begin

Divide database into a chunk of data

For $t = 1: w$

Read d_t

If(t==1)

Build fuzzy system $F_{d\,0}(\mu_0, R_0)$ using d_0

Endif

If(t!=1)

Find accuracy of approximation $\alpha_a(X)$

If $(\alpha_a(X) < T)$

Update fuzzy system to $F_{d\,t-1}(\mu_{t-1}, R_{t-1})$

Endif

Endif

Perform classification of d_t using $F_{d\,t-1}(\mu_{t-1}, R_{t-1})$

Return fuzzy system for classification, $F_{d\,t-1}(\mu_{t-1}, R_{t-1})$

Endfor

End

Figure 3. Pseudo code of HRFuzzy

5. Results and Discussion

This section presents the experimental results of the proposed HRFuzzy system and the quantitative results with the comparison of existing works.

5.1 Dataset Description

The experimentation is performed using two different databases such as Skin Segmentation Data Set and Localization Data which are taken from UC Irvine Machine Learning [19]. *Skin Segmentation Data Set (database 1):* The skin dataset is collected by randomly sampling 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 245057; out of which 50859 is the skin samples and 194198 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 164860.

5.2 Evaluation Metrics

The performance of the proposed HRFuzzy classifier is analyzed using the quantitative parameter, called accuracy. The definition of accuracy is given as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (26)$$

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.

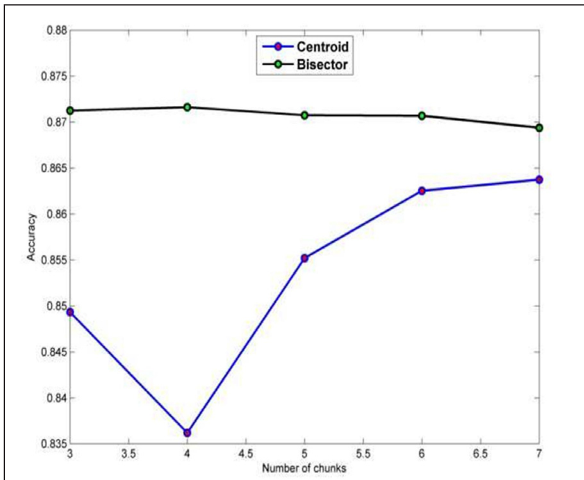
Along with, computational time is also used for comparing the performance of algorithms by calculating running time of the algorithms. Here, we utilize 'tic' and 'tac' variables to measure the computational time of the algorithms through matlab.

5.3 Experimental Setup

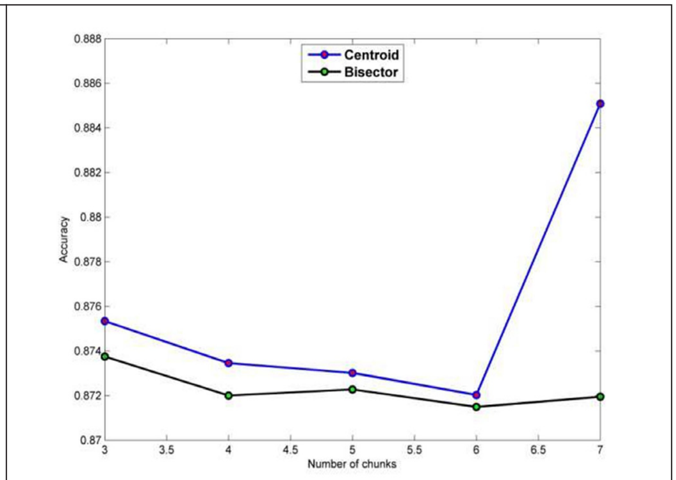
The proposed system is implemented using MATLAB (R2014a) using fuzzy logic toolbox. The system has i5 processor of 2.2GHz CPU clock speed with 4 GB RAM and 64 bit operating system running with Windows 8.1. The parameters to be fixed for the proposed HRFuzzy are number of rules per class *r*, defuzzification method and threshold for change of detection. These parameters are fixed to different range of values and their performance is analyzed to identify the best parametric value for the comparison. The performance is compared with the existing Adaptive k-NN Classifier given in [2].

5.4 Performance Evaluation of the Proposed Algorithm

The performance evaluation of the proposed algorithm is presented in this section. Figure 4.a shows the accuracy graph of different defuzzification method for database 1. From the figure 4.a, we understand that the bisector-based method perform well than the centroid -based method. For the chunk size of 3 ($w = 3$), bisector -based HRFuzzy classifier obtained the accuracy value of 87.2% while comparing with centroid-based method which reached the value of 84.9%. Similarly, centroid and bisector based HRFuzzy classifier reached the value 86.4% and 86.9% for the chunk size of 7. Figure 4.b shows the accuracy graph of different defuzzification method for database 2. This database gives the better performance for the centroid-based fuzzy classifier than the bisector-based classifier. Out of five different size of chunks are experimented, the centroid-based HRFuzzy classifier outperformed in all the five cases.



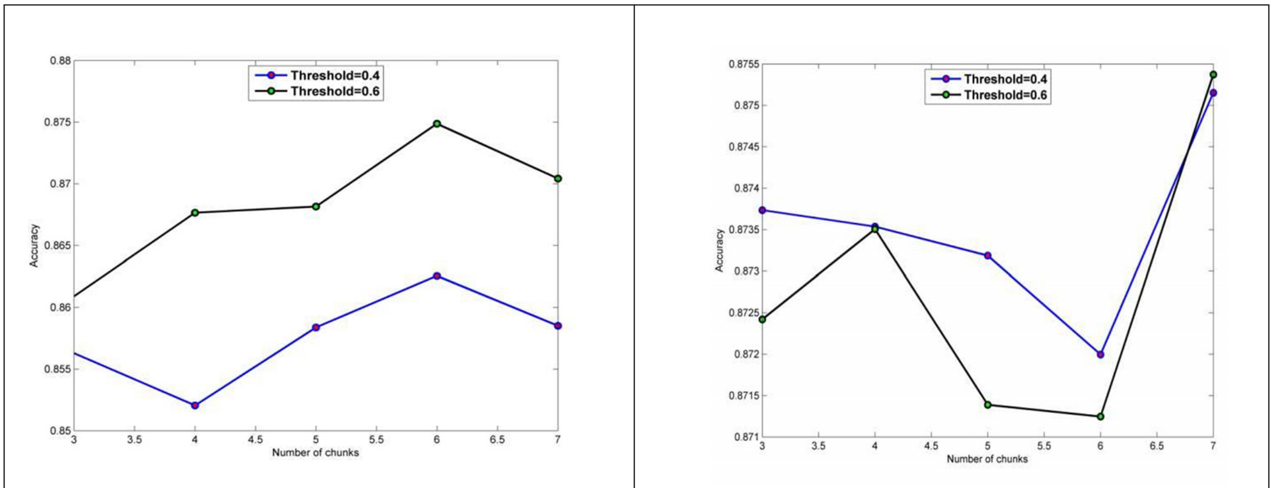
a) database 1



b) database 2

Figure 4. Accuracy for different defuzzification method

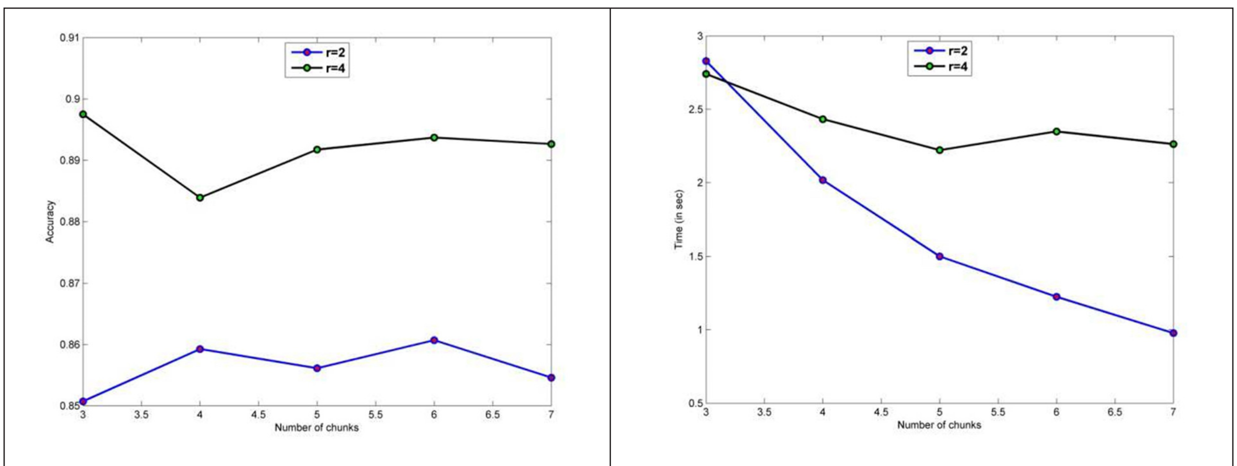
Figure 5.a is plotted after finding the accuracy of the HRFuzzy system for various values of thresholds for change of detection. This figure ensured that higher values of thresholds are desirable because those values have given the better values than the lower values of thresholds. When we fix the threshold is equivalent to 0.6 ($T = 0.6$), we reached the accuracy of 86.1% as compared with accuracy of 85.6% which is obtained for threshold of 0.4. Also, the better performance reached by the threshold (0.6) is 87.5% for the chunk size of six. Figure 5.b shows the accuracy of database 2 for different threshold. Here, better performance is achieved by the threshold of 0.4 which reached the maximum performance of four set of threshold out of five set experimented.



a) database 1	b) database 2

Figure 5. Accuracy for different threshold

Figure 6.a shows the accuracy graph of different number of rules for database 1. From the figure 6.a, we understand that the better performance is reached by the $r = 4$. For the chunk size of 3, the accuracy value of 89.8% is obtained for the rule size of four while comparing with the value of 85%. When we fix the rule size is equivalent to 4 for the chunk size of four, we reached the accuracy of 88.2% as compared with accuracy of 86% which is obtained for rule size of 2. Figure 6.b shows the accuracy graph of different rule size for database 2. The better performance reached by the rule size of 2 is 85.6%, 87.2% and 87.8%. This database gives the better performance for the rule size of 2.



a) database 1	b) database 2

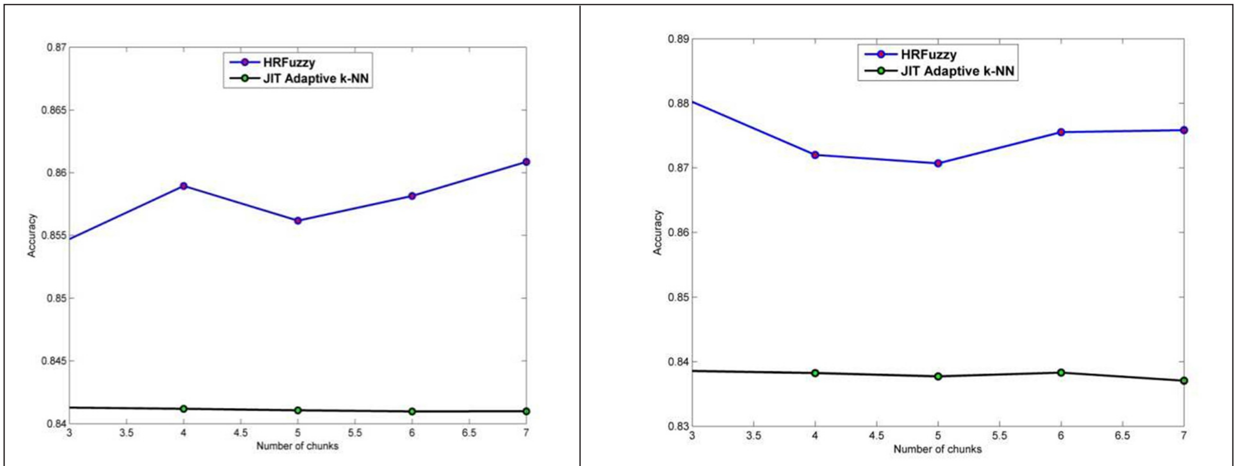
Figure 6. Accuracy for different number of rules

5.5 Comparative Analysis

The comparative analysis of the proposed HRFuzzy classifier is discussed in this section. The existing work taken for the comparison is Adaptive k-NN Classifier given in [2]. Figure 7.a shows the accuracy graph of database 1. From the graph, we understand that the accuracy of the HRFuzzy and Adaptive k-NN are 85.4% and 84.2% for the chunk size of three. Also, for the chunk size of five, the proposed HRFuzzy obtained the value of 85.6% which is higher than the Adaptive k-NN. Similarly, the accuracy of the HRFuzzy and Adaptive k-NN are 86.1% and 84.2% for the chunk size of 7. Figure 7.b shows the performance comparison on database 2. When the chunk size is fixed to 3, the proposed HRFuzzy obtained the accuracy of 88% but the existing

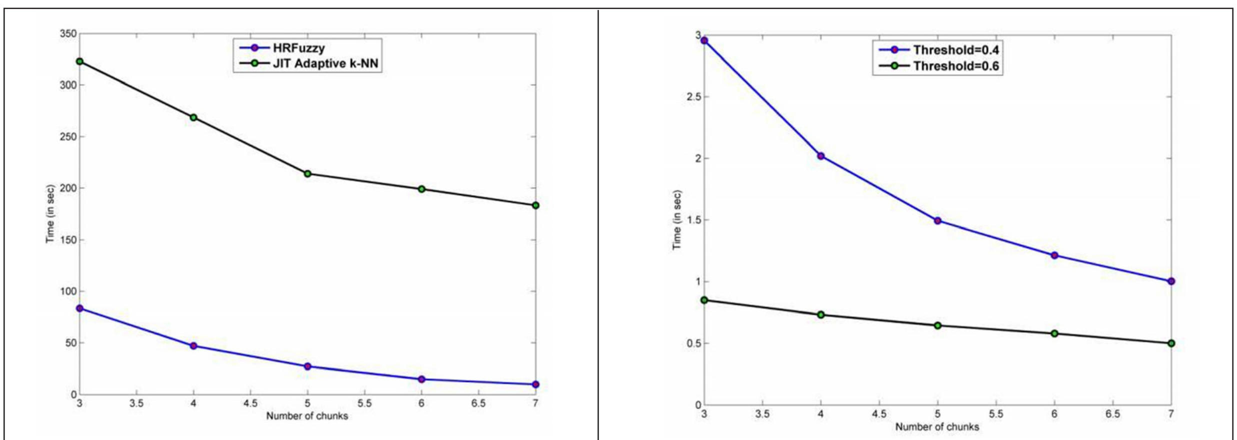
Adaptive k-NN reached the value of 83.8%. For the chunk data size of 6, accuracy of HRFuzzy and Adaptive k-NN are 87.6% and 83.8%.

Figure 8.a shows the comparative performance of the database 2. From the figure, we prove that the proposed HRFuzzy obtained good performance while compared with the Adaptive k-NN. In the chunk data size of 4, the proposed HRFuzzy system take the time of 50sec but the Adaptive k-NN take the computational time of 260sec. The HRFuzzy and Adaptive k-NN take the computational time of 15sec and 180 sec when the chunk data size is 7. Figure 8.b shows the computational requirement of both the algorithms. The minimum time is reached by the HRFuzzy is 2 sec when, the size of chunk is 7. Adaptive k-NN takes the time of about 17sec. By comparing the overall performance for all the two datasets, the proposed HRFuzzy outperformed the Adaptive k-NN by reaching the higher values in accuracy and minimum values in time.



a) database 1	b) database 2

Figure 7. Accuracy



a) database 1	b) database 2

Figure 8. Computational time

6. Conclusion

We have presented a new fuzzy system called, HRFuzzy by combining rough and fuzzy set to classification of evolving data streams. Initially, fuzzy classifier is modeling using the fuzzy membership function and rules which are derived using k-means clustering. Then, the dynamic updating of this model is done using rough set theory and holoentropy function. The concept change is detected by formal approximation of a crisp set in terms of a pair of set through lower and the upper approximation of the original. For the data stream detected with concept drift, the membership function and fuzzy rules are updated using the proposed method. The experimentation of the proposed HRFuzzy is done with two various databases such as Skin Segmentation and Localization Data. The comparison of the proposed HRFuzzy with Adaptive k-NN Classifier is done through accuracy and time. From the comparative analysis, we ensured that the proposed HRFuzzy have obtained good performance in both the metrics. In future, the proposed work can be extended by including learning theories for dynamic updating of classifier model.

References

[1] Mena-Torres., Dayrelis., Aguilar-Ruiz., Jesús, S. (2014). A similarity-based approach for data stream classification, *Expert Systems with Applications*, 41, p 4224–4234, 2014.

[2] Cesare Alippi., Derong Liu., Dongbin Zhao., Li Bu. (2013). Detecting and Reacting to Changes in Sensing Units: The Active Classifier Case, *IEEE Transactions on systems, man, and cybernetics: systems*, 44 (3), 353 - 362.

[3] Zhang, Peng., Zhou, Chuan., Wang, Peng., Gao, Byron., Zhu, Xingquan., Guo, Li. (2015). E-Tree: An Efficient Indexing Structure for Ensemble Models on Data Streams, *IEEE transactions on knowledge and data engineering*, 27, 2, p 461-474, February 2015.

[4] Rutkowski, Leszek., Jaworski, Maciej., Pietruczuk, Lena., Duda, Piotr. (2014). Decision Trees for Mining Data Streams Based on the Gaussian Approximation, *IEEE Transactions on Knowledge and Data Engineering*, 26 (1), 108-119, January 2014.

[5] Brzezinski, Dariusz., Stefanowski, Jerzy. (2014). Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm, *IEEE transactions on neural networks and learning systems*, 25 (1), 81-94, January 2014.

[6] Gomes, João Bártolo., Gaber, Mohamed Medhat., Sousa, Pedro A. C., Menasalvas, Ernestina. (2014). Mining Recurring Concepts in a Dynamic Feature Space, *IEEE Transactions on Neural Networks and Learning Systems*, 25 (1), 95-110, January 2014.

[7] Masud, Mohammad M., Chen, Qing., Khan, Latifur., Aggarwal, Charu, C., Gao, Jing., Han, Jiawei., Srivastava, Ashok ., and Oza, Nikunj, C. (2013). Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams, *IEEE Transactions on Knowledge and Data Engineering*, 25 (7), 1484-1497, July 2013.

[8] Abdulsalam, Hanady., Skillicorn, David, B., Martin, Patrick. (2011). Classification Using Streaming Random Forests, *IEEE Transactions on Knowledge and Data Engineering*, 23 (1), 22-36, January 2011.

[9] Fan, W. (2004). Systematic Data Selection to Mine Concept-Drifting Data Streams, *In: Proceedings ACM SIGKDD 10th Int'l Conf. Knowledge Discovery and Data Mining*, p. 128-137, 2004.

[10]Gao, J., Fan, W., Han, J. (2007). On Appropriate Assumptions to Mine Data Streams, *In*: *Proceedings IEEE Seventh Int'l Conf. Data Mining (ICDM)*, p 143-152, 2007.

[11] Hulten, G., Spencer, L., Domingos, P. (2001). Mining Time-Changing Data Streams, *In: Proceedings ACM SIGKDD Seventh Int'l Conf. Knowledge Discovery and Data Mining*, p. 97-106.

[12] Kolter, J., Maloof. M. (2005). Using Additive Expert Ensembles to Cope with Concept Drift, *In*: Proceedings 22nd *Int'l Conf. Machine Learning* (*ICML*) 449-456.

[13] Wang, H., Fan, W., Yu, P.S., Han, J. (2003). Mining Concept-Drifting Data Streams Using Ensemble Classifiers, *In*: *Proceedings ACM SIGKDD Ninth Int'l Conf. Knowledge Discovery and Data Mining*, p. 226-235, 2003.

[14] Bartolo Gomes, J., Menasalvas, E., Sousa, P. (2010). Tracking recurrent concepts using context,*In*: *Proceedings* 7th Int. Conf. RSCTC, 2010, p. 168–177.

[15] Gama, J., Kosina, P. (2009). Tracking recurring concepts with metalearners, *In*: Proceedings 14th Portuguese Conf. Artif. Intell., October 2009, p. 423.

[16] Katakis, I., Tsoumakas, G., Vlahavas, I. (2005). On the utility of incremental feature selection for the classification of textual data streams, *In:* Advances in Informatics. New York, NY, USA: *Springer-Verlag*, 2005, p 338–348

[17] Yang, Y., Wu, X., Zhu, X. (2006). Mining in anticipation for concept change: Proactive-reactive prediction in data streams, *Data Mining Knowl. Discovery*, 13 (3), 261–289, 2006.

[18] Pawlak, Zdzisaw. (1982). Rough sets, *International Journal of Parallel Programming*, 11 (5), 341–356.

[19] UC Irvine Machine Learning Repository from "http://archive.ics.uci.edu/ml/datasets.html".

[20] Ross, G. J., Tasoulis, D. K., Adams, N. M. (2012). *Nonparametric monitoring of data streams for changes in location and scale, Technometr*., 53(4), 379–389.

[21] Zhu, X., Zhang, P., Lin, X., Shi, Y. (2010). Active Learning from Stream Data Using Optimal Weight Classifier Ensemble, *IEEE Trans. System, Man, Cybernetics,* Part B: *Cybernetics*, 40 (4), 1-15, December 2010.

[22] Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavalda, R. (2009). New Ensemble Methods for Evolving Data Streams, *In:* Proceedings 15th *ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining* (KDD), 2009.

[23] Masud, M., Gao, J., Khan, L., Han, J., Thuraisingham, B. (2011). Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints, *IEEE Trans. Knowledge and Data Eng*., 23 (6), 859-874, June 2011.

[24] Wu, Shu., Wang, Shengrui . (2013). Information-Theoretic Outlier Detection for Large-Scale Categorical Data, *IEEE Transactions on Knowledge and Data Engineering,* 25 (3), March 2013.

[25] Zadeh, L. (1992). Fuzzy sets. *In:* Fuzzy models for pattern recognition: Methods that search for structures in data, NY: *IEEE Press*, 1992.

[26] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *In*: Proceedings 5th Berkeley Symp. Math. Stat. Probab., L. M. L. Cam and J. Neyman, Eds. Berkeley, CA: Univ. California Press, 1967, vol. 1.