International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 38589-38597

The Mapreduce Based MRMPCME Algorithm for Big Stream Data

G.Somasekhar,
Research Scholar, SCSE, VIT University,
Vellore,TamilNadu,India,
Ph: +919908219507,Email:gidd.somasekhar2014@vit.ac.in

Dr.K.Karthikeyan,
Associate Professor, SAS, VIT University,
Vellore,TamilNadu,India,
Ph: +919944943681,Email:k.karthikeyan@vit.ac.in.

Abstract- Stream data processing and stream data mining are the hot research topics today. Many solutions are given by the research community on various problems of stream data from time to time. Single model(incremental) classification, single model(incremental) clustering, Ensemble classification, and Ensemble clustering are the basic techniques which had been received much interest from researchers. Several single model(single partition single chunk) techniques are proposed for stream data classification[15,16,17,18]. The single model techniques need complex operations to modify the internal structure of the model and use most recent data to update the model , forgetting a lot of historical data. Ensemble methods are the other side of the coin which overcome the drawbacks of single model methods. Though the ensemble methods are giving more classification accuracy than the former, they ignore some amount of historical data while pruning the weakest classifier in the ensemble periodically. A multi partition multi chunk ensemble classification technique(MPC approach) was first proposed by Mohammad M. Masud et. al.[1] to gain maximum classification accuracy without ignoring the historical data of even a single data chunk. This technique has some limitations. It's main focus is only on classification accuracy. The other problems like classification cost, time eating, and scalability are ignored. These are overcome by the proposed MRMPCME(mapreduce based multi partition multi chunk mixed ensemble) approach. The classifier and cluster ensemble technique[2] is combined with the original MPC approach and applied mapreduce in addition. The solution saves the execution time, reduces the classification cost and solves the scalability problem. It is proved to be a better technique when compared to the state-of-the-art techniques of stream data classification and more suitable for big stream data applications.

Keywords: Ensemble, Classification, Mapreduce, Multi partition, Multi chunk, Clustering, Stream data, Big stream data, Stream data mining.

I. Introduction

We are in the era of big data. At present the data generated by big data sources per day is in peta bytes range and it may reach exa bytes or zetta bytes in near future. The wireless sensor data, radio frequency identification(RFID) data, and web traffic data are the best examples. These data arrive continuously and rapidly which form a new class of data called "big stream data". The classification of big stream data became a challenging task because of the following problems.

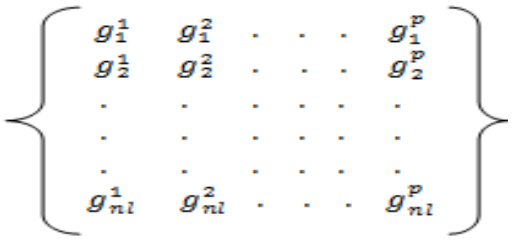
i) It is highly labor-intensive. ii) It is time eating. iii) Scalability problem arises. iv) Complex characteristics of stream data such as infinite length, evolving nature and concept drift become major hurdles to classification.

Various solutions are proposed to solve the issues of concept drifting data streams(the last problem). One of the main issues in mining concept-drifting data streams is to select the appropriate train data set to learn the evolving concept. The following are some techniques proposed by researchers to handle concept drifting data streams. i)The technique which selects and stores the training data that are most consistent with the current concept [25]. ii)The technique which updates the existing classification model when new data appear, such as the Very Fast Decision Tree(VFDT)[15] approach. iii)Other single model (incremental) approaches[16, 17, 18]. iv)The ensemble approach which uses an ensemble of classifiers and updates the ensemble every time new data appear[4]. For handling unexpected changes and concept drifts, the ensemble classifier is proven to be the best solution[4,26]. All the ensemble approaches except the multi partition multi chunk approach(MPC approach)[1] use a single partition and a single chunk at a time. The multi partition multi chunk approach(MPC approach)[1] improves the stream data classification accuracy significantly by using multiple partitions and multiple chunks at a time. But the other problems(high labor cost for classification, time eating for classification, and scalability)are not focused in many solutions including MPC. The proposed MRMPCME algorithm gives an efficient solution in this paper addressing these problems by getting the advantage from the combination of the following techniques. i)Mapreduce ii)The MPC approach iii)The classifier and cluster ensemble technique[2].

The classifier and cluster ensemble technique[2] gives higher reduction in classification labor cost and it also reduces the time taken for labeling process. By using the big data technique called mapreduce, the scalability problem can be solved. The MPC approach maintains the accuracy. As the MRMPCME algorithm is the combination of all the above three techniques, it gains all the respective advantages.

In the MPC approach, there are three controlling parameters : v, r, and k. The parameter 'v' determines the number of partitions (v=1 means single-partition ensemble), the parameter 'r' determines the number of chunks (r= 1 means single chunk ensemble), and the parameter 'k'

controls the ensemble size. The ensemble consists of k*v models. In the MRMPCME approach, 'v' is replaced by 'p'(to differentiate between 'v' classifiers in MPC and 'p' cluster models in MRMPCME).This ensemble is updated whenever a new data chunk is labeled. The labeling process may be either classification or the sequence of steps including label propagation and cluster internal structure information propagation[2]. As the classification process eats time, while very few of the data chunks are being classified, huge number of unlabeled data chunks complete their labeling process through the label propagation and cluster internal structure information propagation steps. The initial ensemble E_{k*p} is the set of best k*p models of all the models from m_1 to m_n. This selection of best models is based on the measured consistency with respect to the up-to-date model m_n. The top most k*p models having higher consistency are selected. To achieve the better accuracy than the usual ensemble approaches, the most recently labeled 'r' consecutive data chunks(through the steps label propagation and cluster internal structure information propagation) are merged, shuffled and divided into 'p' number of equally distributed partitions. Each labeled partition d_i contains equally distributed parts of the information of all the recent 'r' data chunks(p-fold partitioning) and it is modeled by taking d_i as the test set and D_p-{d_i} as the training set where D_p is the entire data set containing all the data in 'p' partitions. By comparing the new label obtained by this process with the existing label information, the consistency of the clustering model λ_i is determined. The graph G containing information about all the models from λ_1 to λ_p is used in this process. The generation of graph G is discussed in [2] and section IV. E^n is the set of 'p' models formed in this way(where $E^n=\{\lambda_1,\lambda_2,\ldots,\lambda_p\}$).Each model λ_i is the representative of all the recent 'p' partitions.



Graph 'G'

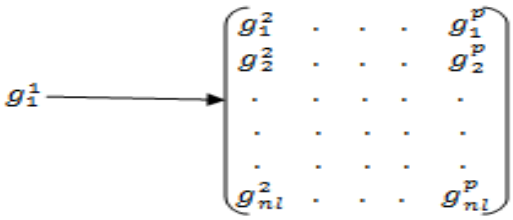


Fig.1. The content of graph 'G' and group g_1^1 of partition'1' having similarity comparisons with remaining groups in other partitions.(Subscript of group g_1^1 is existing label number and superscript of g_1^1 is the partition number. Similarly each group has it's own label number and partition number).

So pruning of any model formed in this way does not mean ignoring a whole data chunk. As the number of recent chunks 'r', the number of partitions 'p' and the number of labels 'nl' are large in real time, the size of graph G is large and the problem is more closer to a big data scenario. The scenario is as shown below.

From fig.1, total number of similarity comparison pairs formed with group g_1^1 = (p-1)*nl ---- (1)

Total number of similarity comparison pairs formed with each partition = nl*(p-1)*nl---- (2)

Total number of similarity comparison pairs formed for 'p' partitions = p*nl*(p-1)*nl = $2*nl^2*pC_2$ ---- (3)

So, total number of similarity pair comparisons will become very large if r,p and nl become large.

The new mixed ensemble E_{new} is the set of best k*p models of the set (E_{k*p} U E^n).The total number of models in the ensemble is always kept constant. The proposed MRMPCME is compared with MPC and other ensemble approaches. The results prove that the proposed approach is better than the MPC and other state-of-the-art ensemble approaches.

II. Related Work

Many solutions were given by researchers to solve the problem of big stream data classification. The basic approaches are, (i)Single model classification (ii)Ensemble classification. Various single model solutions are proposed in [15,16,17,18].A comparative study between single model(incremental) and ensemble learning on data streams was done by Wenyu Zang, Peng Zhang, Chuan Zhou and Li Guo[3].Due to their complexity and less classification accuracy, the single model methods are replaced by ensemble methods now a days. The alternatives to single model in a non-streaming environment are discussed in [19].several ensemble techniques for data stream mining have been proposed in [20,21,22]. A review on real time data stream classification in various concept drifting scenarios was done by Ms.Priyanka B. Dongre and Dr.Latesh G.Malik[14].The solution of ensemble classification accuracy. Several Mapreduce based classification solutions were given by Latifur Khan, Ahsanul Haque and others[10,11,12,13].

III. Motivation

Big stream data classification has become a challenging task for real time applications like credit card fraud detection, target marketing, network intrusion detection and so on. In this paper big stream data classification and clustering are focused. Many single partition single chunk approaches are proposed by researchers in [15,16,17,18] taking single partition and single data chunk at a time for classification. In [1],Mohammad M. Masud and others proposed a multi partition multi chunk ensemble technique based on new ensemble formation for last 'r' labeled data chunks. This

technique focused on the accuracy of classification and there was no discussion about classification cost, scalability and speed-up. The above three goals are achieved through the proposed solution. The mapreduce based stream data classification approaches[10,11,12,13], the MPC approach[1] and semi-supervised learning approaches for stream data classification [2,23,24] motivated us to develop the algorithm.

IV. Problem Statement

When the big stream data arrives more rapidly, labeling of the data instances becomes much difficult, time consuming and cost consuming. One idea is adding sufficient number of processors and computing labels in parallel. But this is not economical always. Furthermore, the big stream data flow rate may be so high that we can not increase the processing capability at a time. So another alternate idea is using clustering methods which is more economical and practical compared to the first solution. We proposed a mapreduce based solution using the above idea which focuses on scalability and speed-up issues of the big stream data classification problem. A multi partition and multi data chunk approach(MPC) is considered to improve the accuracy of big stream data classification. This approach is combined with clustering methods to meet the needs of cost reduction and speed-up.The below figure(fig.2) depicts the idea clearly.

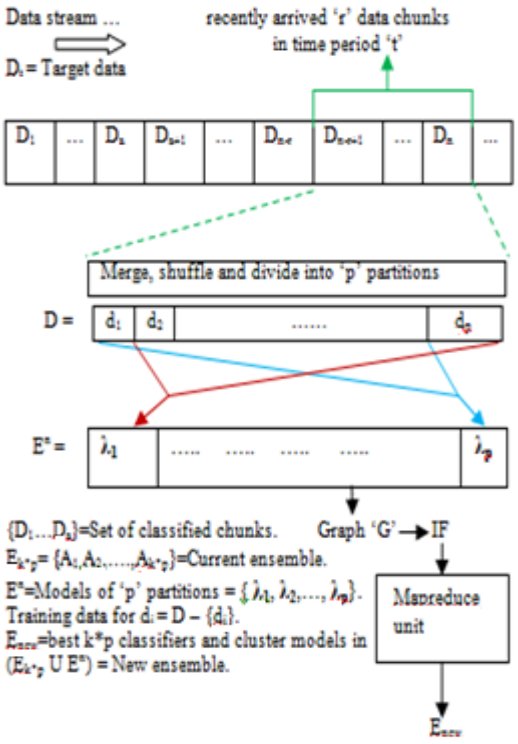


Fig.2.The MRMPCME approach.

Target data=D_t= Set of classified data chunks+ Set of clustered data chunks+ Set of recently labeled 'r' consecutive data chunks =
$\{D_1+D_2+\ldots+D_a\}+\{D_{a+1}+D_{a+2}+\ldots+D_{n-r}\}+\{D_{n-r+1}+D_{n-r+2}+\ldots+D_n\}=\{D_1+D_2+\ldots+D_a\}+\{D_{a+1}+D_{a+2}+\ldots+D_{n-r}\}+\{d_1+d_2+\ldots\ldots+d_p\}$.
{after merging and shuffling of recently arrived 'r' data chunks and dividing them into 'p' partitions}.
$D_1+D_2+\ldots+D_a$ = Set of 'a' classified data chunks,
$D_{a+1}+D_{a+2}+\ldots+D_{n-r}$ = Set of 'c' clustered data chunks.{where c = n-r-a},
$D_{n-r+1}+D_{n-r+2}+\ldots+D_n$ = Set of 'r' recently labeled consecutive data chunks.
$m_1,m_2,\ldots,m_n$ are the models of the data chunks from D_1 to D_n.(Note: For the data chunks other than 'a' classified data chunks, the labeling process is the sequence of steps called label propagation and cluster internal structure information propagation).
Current classifier and cluster ensemble=E_{k*p}=best k*p models in $\{m_1,m_2,\ldots,m_n\}=\{A_1,A_2,\ldots,A_{k*p}\}$.
p' newly trained models of 'p' partitions=$E^n=\{\lambda_1, \lambda_2,\ldots,\lambda_p\}$.
New classifier and cluster ensemble(mixed ensemble)=E_{new}=best k*p classifiers and cluster models in (E_{k*p} U E^n).

The number of classifiers and the number of cluster models in the ensembles E_{k*p} and E_{new} are in the ratio a:c+r. The problem is to determine the new ensemble E_{new} with less cost consumption and less time consumption added with the advantage of scalability.

V. Problem Solving

Assume a big stream of data 'S' containing an infinite number of data records (x_i, y_i), where $x_i \in R_d$ denotes an instance containing d-dimensional attributes, and $y_i \in Y = \{c_1,\ldots, c_{nl}\}$ represents its class label (Note that only a small portion of data records are labeled and actually contain the labeling information y_i).

Let us assume all the data records those are labeled by classification are divided into 'a' number of consecutive data chunks(where a≥1).Since the classification is time consuming, in real time, most of the data chunks are unlabeled and a very few number of initial data chunks (Say 'a') are labeled by classification process. Of all the unlabeled data chunks, 'r' number of data chunks are recent. Usually each classifier is determined from each data chunk. Since determination of classifier is time consuming and costs more labor, the alternate choice is clustering. So very small number of data chunks are classified and large number of data chunks are clustered to reduce the labor cost and computation time. Label propagation and cluster information propagation are used to determine the labels for all the clusters as accurate as possible. Finally an ensemble of 'k*p' models is formed. This ensemble is very useful in the decision making process.

In the usual approach, each time a new classifier is determined from single data chunk and no partitions are done. If this new classifier's accuracy is more, it replaces the weakest classifier in the ensemble. A different ensemble formation technique(MPC) using multiple partitions and multiple chunks[1] is considered in the proposed approach.

Rather than usual single partition-single data chunk approach, a multi partitioning multi data chunk approach is applied on the training data to improve the accuracy of labeling process. In the rapidly arrived 'n' number of consecutive data chunks, a small number, say 'a' number of data chunks are classified and all the remaining 'c+r'(where n-a=c+r) data chunks are clustered and labeled through label propagation and cluster internal structure information propagation steps. The models of the remaining 'c+r' data chunks are termed as cluster models here to differentiate them from 'a' classifiers. In the 'c+r' number of remaining data chunks, the 'r' number of data chunks are recent and clustered. An initial classifier and cluster ensemble is formed with top most consistent k*p models by using Eq(1).Then the recent 'r' number of data chunks are divided into 'p' equally distributed partitions. The proposed algorithms are applied on these 'p' number of partitions to handle the concept drift.

In real time scenarios, as most of the stream data is unlabeled, instead of the classification accuracies of the models, the consistencies of all the classifiers and cluster models in the ensemble with the up-to-date cluster model are considered. The classifier or cluster model with more consistency replaces the classifier or cluster model with least consistency. Let m_n be the up-to-date model, then for a base model $m_i (1 \leq i < n)$, its weight $W(m_i)$can be computed using,

$W(m_i)=Sim(m_i,m_n)/z$ ------- Eq.(1) (where $z = \sum_{i=1}^{n} Sim(m_i,m_n)$,and z is called a regularization factor) .

The top k*p models(classifiers and cluster models are in the ratio a:c+r) having best consistency values measured with the up-to-date model m_n are formed as a mixed ensemble, say E_{k*p}, pruning the poorly consistent classifier or cluster model each time a new model is added.

According to the proposed approach, recently labeled 'r' consecutive chunks are taken. They are merged shuffled and divided into equally distributed 'p' partitions by using p-fold partitioning.

The models of the 'p' number of partitions are named from $\boldsymbol{\lambda_1}$ to $\boldsymbol{\lambda_p}$ (where $E^n=\{\boldsymbol{\lambda_1},\boldsymbol{\lambda_2},\ldots,\boldsymbol{\lambda_p}\}$, and D_p is the entire set of data points under 'p' number of partitions).Each i^{th} partition d_i is modeled considering partition 'd_i' as the test set, and D_p-d_i as the training set and it's consistency is measured by using step 6 in algorithm 1.

New Ensemble=E_{new}=Best consistent (k*p) models of $\{E_{k*p} \cup E^n\}$ with classifiers and clustering models in the ratio a:c+r.

Each model related to any partition in the 'p' number of partitions is the representative of all the 'r' recent data chunks. Though some models from a chunk may have been removed, other models from that chunk may still remain in the ensemble. Whereas, in the approach of Wang et al.[4], removal of a classifier means total removal of the knowledge obtained from one whole chunk. In addition to this, The mapreduce technique is applied to this approach to serve the purpose of scalability in big stream data applications. The proposed algorithms use graph G = {V,e} as the basic input which is generated by the procedure given in [2] where each vertex, 'V' in graph 'G' is a group formed by a model $\boldsymbol{\lambda_i}$ from the ensemble E^n and each edge 'e' in the graph G is the similarity between any two groups in 'G' measured by the Euclidian distance between any two groups using Eq(2).

$Sim_g(g_k^i,g_h^j) = d^{-1}(g_k^i,g_h^j) = 1 / \|g_k^i - g_h^j\|_2^2$ (where i≠j, $1 \leq k \leq$ nl, and $1 \leq h \leq$ nl) ------ Eq.(2)

(Note: nl=number of existing labels, i and j are two different partition numbers).

The proposed algorithms including mapper and reducer processes are mentioned below.

Algorithm1. MapReduce based multi partition multi chunk mixed ensemble(MRMPCME) approach

MRMPCME(D_t,a,c,r,n,E_{k*p},G,D_n,m_n,p,nl,Y_{old})

Input: D_t= Target big stream data set={ $D_{1\ldots}D_a$}+{ $D_{a+1\ldots}D_{n-r}$}+{$D_{n-r+1\ldots}D_n$}
{ $D_{1\ldots}D_a$}=set of classified data chunks , and
{ $D_{a+1\ldots}D_{n-r}$}= set of clustered data chunks. {a <<< n}.
{$D_{n-r+1\ldots}D_n$}= set of 'r' recent, consecutive and clustered data chunks.
a =number of classified chunks or number of classifiers.
c+r=number of clustered data chunks through label propagation and cluster internal structure information propagation.
r=number of recently labeled data chunks to be merged,shuffled and divided into 'p' partitions.
n=Total number of data chunks.
nl=Total number of existing labels.
E_{k*p}=Initial mixed ensemble of k*p models.
G=The graph representation for all the groups in all the 'p' number of partitions.
D_n=Up-to-date data chunk.
m_n= Model of up-to-date data chunk D_n.
Y_{old}= Array of the existent group information vectors related to all partitions ={$y^1,y^2,\ldots,y^p$} where y^i is the group information vector pertaining to partition 'd_i'.
p = The total number of partitions made by dividing 'r' consecutive data chunks.

Output:
E_{new}=New mixed ensemble.

1: E_{k*p}=Best k*p models from all the models including recent model m_n with the consistencies or weights measured using Eq(1).
2: Merge, shuffle and divide 'r' recently labeled consecutive data chunks into 'p' partitions by p-fold partitioning.
3: IF = WRITE(G,1,p,nl);
4: $\bar{Y}_{new}$ =MR-MULTIPARTITION-LABEL(IF,nl,$\bar{Y}$,f,v);
5: **for** f=1,…,p **do**
6: $W(\boldsymbol{\lambda_f})$ = Sim($\bar{y}^f$,y^f) , where $\bar{y}^f$ is the vector of all aggregate(f_k')s related to partition d_f obtained from $\bar{Y}_{new}$ and y^f is the existent group information vector related to partition d_f obtained from Y_{old}.
7: **end for.**
8: E_{new} = Best consistent (k*p) models of $\{E_{k*p} \cup E^n\}$ with classifiers and cluster models in the ratio a:c+r, where consistencies of partitions or chunks are W values(weights) in the algorithm as mentioned in the above equation in the step 6.
{ Here | y^f|=| $\bar{y}^f$|=nl in the step 6 and IF is the input file to the mapreduce job}.

Algorithm2.Pseudocode for mapper of MR-MULTIPARTITION-LABEL( )

MAP-MULTIPARTITION-LABEL(IF,nl, $\bar{Y}$,f,A)

Input:
IF=Input file containing similarity pairs each of the form (g_x, g_y) where g_x and g_y are group information vectors related to any two different partitions.
$\bar{Y}$=Set of all g_k^f values initialized to '0'.
nl=number of labels for each classifier or cluster model.
f=Serial number of a partition.
A=Total number of all partitions.

Output: $\bar{Y}$ = Set of all modified g_k^f values.

1: **for** f ∈ {1,…,A} and k ∈ {1,…,nl} **do**
2: **for** i ∈ {1,…,A}-{f} and h ∈ {1,…,nl} **do**
3: $g_k^f = argmax(g_k^f + \text{Sim}_{y \in Y}(g_k^f, g_h^i)\text{Label}((g_h^i))$;
(where, g_k^f **is** group/cluster of partition 'd$_f$', g_h^i **is** group/cluster of any partition other than partition 'd$_f$' , 1≤h≤nl,1≤k≤nl, and i ∈ {1,…,A}-{f}).
4: **end for;**
5: **end for;**
{ Here, (g_k^f, g_h^i) is a line(or a tuple) in the input file IF and the argument of the maximum (abbreviated **arg max** or **argmax**) is the set of points of the given argument for which the given function attains its maximum value. }.

Algorithm 3. Pseudo code for reducer of MR-MULTIPARTITION-LABEL( )

REDUCE-MULTIPARTITION-LABEL(Y_s)

Input:

Y_s=Union of all $\bar{Y}$s.(Where each $\bar{Y}$ is the output of each mapper)

Output: Finalset $\bar{Y}_{new}$.

1: **for** f ∈ {1,…,A} **do**
2: aggregate(g_k^f) = argmax(Union of all g_k^f s related to same partition 'd$_f$' where 1 ≤ k≤ nl)
3: **end for.**
4: $\bar{Y}_{new}$ = Set of all aggregate(g_k^f)s. (where 1≤ f≤ A and $|\bar{Y}_{new}|$ = nl.A)
5: return $\bar{Y}_{new}$.

Algorithm 4.Pseudo code for writing input file for mapreduce job.

WRITE(G,a,b,nl)

Input:
Graph G of classifiers/clusters.
a= starting partition.
b=ending partition.
nl=number of existent labels.

Output: Input file IF.(used as an input for the mapreduce job).

1: **for** f=a…b **do**
2: **for** k=1…nl **do**
4: **for** m=a…b **do**
5: **if** m=f **then**
6: **go to** 4;
7: **for** s=1…nl **do**
8: Write each line of group information (g_k^f,g_s^m) in the input file IF.
9: **end for;**
10: **end for;**
11: **end for;**
12: **end for;**
13: return IF.

VI. Highlights

Our contributions are highlighted as follows.

1)Reduction in time complexity:

From [1], it is observed that the time complexity of MPC approach is O(n.(Ks+f(rs))),where 's' is the size of one data chunk, 'n' is the total number of data chunks, f(x) is the time to build a classifier on a training data of size 'x'. Let us assume f^l(x) is the time to build a cluster model on a training data of size 'x' using mapreduce. While the MPC approach is taking ' rp' times higher running time than that of Wang et al[4],getting significant error reduction, the actual running time of MRMPCME is 'μ' times lesser than that of MPC[1]without any compromise on the accuracy. The time complexity of the proposed approach becomes O(n.(Ks+f^l(rs))) (According to MRMPCME approach).
= n.O(Ks+f^l(rs)) where r = number of recently labeled chunks to be merged, shuffled and divided into 'p' partitions.

As the MRMPCME uses mapreduce and semi supervised learning approaches, it's actual running time takes 'μ' times lesser running time than that of MPC.
Where, T_z=Total time taken for label propagation + Total time taken for cluster internal structure information propagation+ Total time taken for mapreduce.=T_l+T_{cis}+T_{MR}.
T_c = Estimated classification time for all the recent data chunks, if they were classified= (Average classification time of 'a' classified chunks)* r .
μ = T_c/T_z (where T_z<< T_c)
Hence Time complexity is reduced by 'μ' times than that of MPC.

2)Reduction in classification labor cost: As huge number of data chunks are labeled by label propagation and cluster information propagation, the classification labor cost is reduced a lot compared to MPC.

3)Solving the scalability problem: For a large number of 'p' or 'nl' the result is obtained in time because of the use of big data technique called mapreduce. This solves the scalability issue.

The above contributions, and the results in section VI show that the proposed MRMPCME is better when compared to MPC and other state-of-the-art ensemble approaches as the latter failed in solving the above three

problems(i.e. Time eating by classification, classification labor cost, and scalability issue).

VII. Results and Comparative Analysis

Disadvantages of the MPC approach and other ensemble approaches:

a)Not suitable to big stream data applications: As the MPC approach does not use any big data technique like mapreduce, it is not suitable for big stream data applications.

b)Classification cost: As the MPC approach only depends on classification, it is cost-consuming process.

c)Time consumption: More time is needed for classification in the MPC approach and other ensemble approaches.

d) In most of the ensemble approaches, classifier removal from the ensemble means removal of the knowledge from whole data chunk.

e)No scalability: If the data chunk size goes very large or the variable 'r'(number of multiple chunks taken each time in the MPC approach) increases to a large number, then the MPC approach is not scalable. Even other ensemble approaches are also not scalable.

Advantages of the proposed approach :

a)More suitable for big stream data applications: Since, MRMPCME uses big data technique like mapreduce, it is most suitable for big stream data applications.

b)Less classification cost: As the number of data chunks classified are very limited in number, classification cost is reduced to a greater extent compared to MPC approach.

c)Less time consuming: As the label propagation, the cluster information propagation, and the mapreduce processes take less time compared to traditional processes, the actual running time is reduced to a greater extent.

d)Though some classifiers are removed from the ensemble, other classifiers of the particular chunk still remain in the ensemble. So no loss of the knowledge pertaining to that data chunk.

e)Solves the scalability issue: The semi supervised learning processes and the mapreduce solve the problem of scalability.

TABLE.1.Execution time Comparison for synthetic data

Cluster size (in number of data points)	MPC (Execution time in seconds)	MRMPCME (Execution time in seconds)
250	130	56
500	135	59
750	138	61
1000	140	62

The execution times are measured for different sized chunks,for both MPC and MRMPCME approaches as shown in tables.1&2 for both synthetic and real data sets. For the results shown in fig.3, fig.4, fig.5 and fig.6 the parameter values are fixed(i.e. K=8,p=5 and r=2).

TABLE.2.Execution time comparison for real data

Cluster size (in minutes)	MPC (Execution time in seconds)	MRMPCME (Execution time in seconds)
30	4.8	2.3
60	5.8	2.8
90	5.9	3.0
120	6.0	3.1

It is observed that MRMPCME takes lesser execution time than MPC approach, maintaining the maximum accuracy(minimum error %) as MPC. The pseudo distributed mode Hadoop 2.2.0and ubuntu 13.10 operating system environment is used for mapreduce jobs. The comparison graphs are shown below in fig .3, fig.4, fig.5 and fig.6 for both synthetic and real data sets. As the semi supervised learning methods(label propagation and cluster internal structure information propagation)are used, the classification labor cost is reduced a lot. For example, if the percentage of classified chunks is 30%,then the remaining chunks are subjected to semi supervised learning reducing the classification cost to 60% approximately.(where, 10% is the overhead cost for semi supervised learning).

For the constant chunk size(parameter 's') of 750 data points, execution times are measured for MPC and MRMPCME taking different 'r' values, different 'p' value

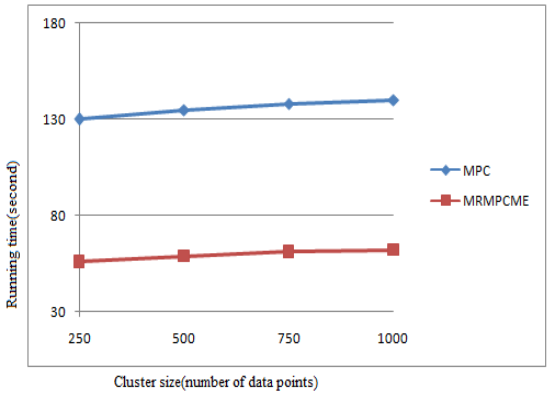


Fig.3.Execution times comparison of MPC and MRMPCME approaches for synthetic data.

-s and different 'nl' values each time keeping the other parameters constant. From fig.7, fig.8 and fig.9, it is observed that the performance of MPC degrades when 'r' 'p' and'nl' grow, going beyond time limits and we can conclude that MRMPCME has the ability to handle the stream data classification problem when the parameters 'r', 'p' and 'nl' grow. So scalability issue is solved. The above results prove that the proposed approach works far better than the MPC approach and other state-of-the-art big stream data classification approaches.

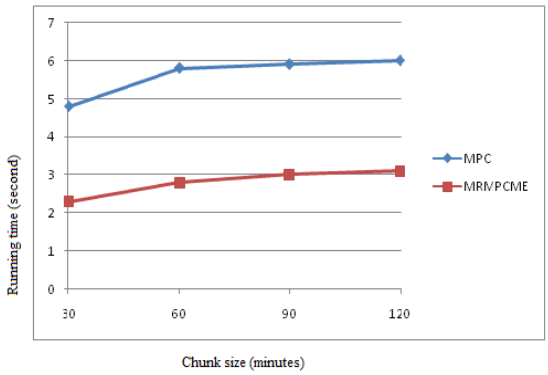


Fig.4.Execution times comparison of MPC and MRMPCME approaches for real data.

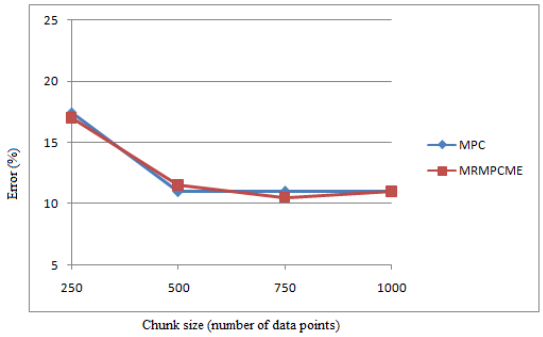


Fig.5.Error vs chunk size comparison of MPC and MR - MPCME approaches for synthetic data

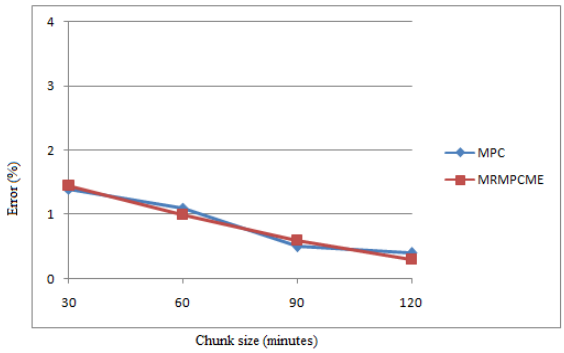


Fig.6.Error vs chunksize comparisonof MPC and MRMPCME approaches for real data.

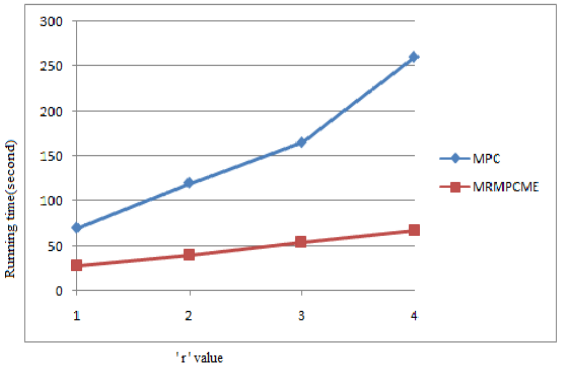


Fig.7.Execution times comparison of MPC and MRMPCME approaches for different ‘r’ values, keeping p, s, and nl constant.

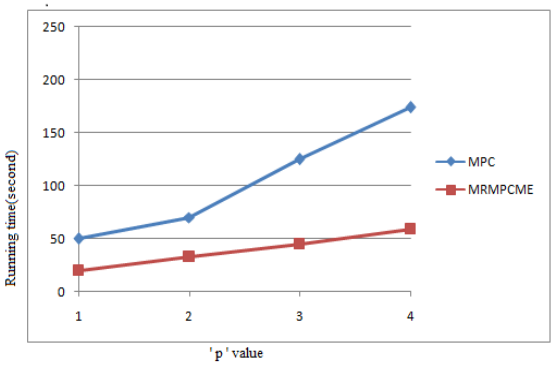


Fig.8.Execution times comparison of MPC and MRMPCME approaches for different ‘p’ values, keeping r, s, and nl constant.

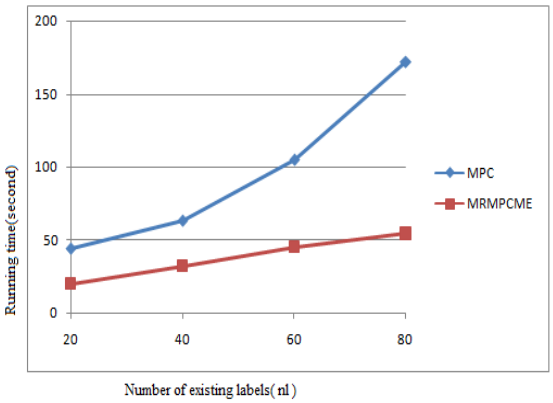


Fig.9.Execution times comparison of MPC and MRMPCME approaches for different ‘nl’ values, keeping r, p, and s constant.

VIII. Conclusion

Many solutions are proposed for stream data classification using single chunk and single partition. MPC is recent technique that uses multiple partitions and multiple chunks without focusing on classification cost, execution time and scalability issues. But it maintains maximum accuracy compared to other state-of-the art approaches. The proposed MRMPCME approach solves the above three problems maintaining the maximum accuracy and is more suitable for concept drifting and evolving big data streams in real time, when number of existing labels and number of partitions are large. It is proved that the MRMPCME performs better than MPC and other state-of-the-art approaches. In the future, we would focus on other real time big data scenarios in various fields.

References

[1] Mohammad M. Masud, Jing Gao, Latifur Khan, JiaweiHan, and BhavaniThuraisingham,2009, “A Multi-partition Multi-chunk Ensemble Technique to Classify Concept-Drifting DataStreams*”,13th Pacific-Asia Conference, PAKDD 2009, Springer-Verlag Berlin Heidelberg*, pp. 363–375.

[2] Peng Zhang, Xingquan Zhu, Jianlong Tan and Li Guo ,2010,"Classifier and Cluster Ensembles for Mining Concept Drifting Data Streams",*10th IEEE International Conference on Data Mining ,IEEE*, pp. 1175 – 1180.

[3] WenyuZang, Peng Zhang, Chuan ZhouandLi Guo, ,2014,"Comparative study between incremental and ensemble learning on data streams: Case study", *Journal Of Big Data 2014,1:5,Springer*.

[4] Haixun Wang , Wei Fan , Philip S. Yu , and Jiawei Han,2003,"Mining concept-drifting data streams using Ensemble classifiers", *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, pp. 226-235.

[5] Mohammad M. Masud, Jing Gaoz, Latifur Khan, Jiawei Han, and BhavaniThuraisingham,2010, "Classification and Novel Class Detection in Data Streams with Active Mining", *PAKDD'10 Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II, Springer-Verlag Berlin, Heidelberg*, pp. 311-324.

[6] Mohammad M. Masud, Qing Chen, Jing Gao, Latifur Khan, Jiawei Han, and BhavaniThuraisingham ,2010,"Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space", *ECML PKDD'10 Proceedings of the2010 European conference on Machine learning and knowledge discovery in databases: Part II, Springer-Verlag, Berlin, Heidelberg* , pp.337-352.

[7] Mohammad M. Masud, Jing Gao,Latifur Khan, Jiawei Han, and BhavaniThuraisingham,2009, "Integrating Novel Class Detection with Classification for Concept-Drifting Data Streams", *ECML PKDD '09 Proceedings of the European Conference on Machine Learning and Knowledge Discovery in databases: Part II, Springer-Verlag Berlin, Heidelberg*, pp. 79 – 94.

[8] Mohammad M. Masud · Clay Woolam · Jing Gao ·Latifur Khan · Jiawei Han · Kevin W. Hamlen ·and Nikunj C. Oza,2012,"Facing the reality of data stream classification: coping with scarcity of labeled data "*Knowledge and Information Systems, Volume 33,Issue 1, Springer-Verlag*,pp. 213-244.

[9] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and BhavaniThuraisingham,2011, "Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints", *IEEE Trans-actions on Knowledge and Data Engineering, Volume 23 Issue 6*, pp.859-874.

[10] AhsanulHaque, Brandon Parker and Latifur Khan,2013," Labeling Instances in Evolving Data Streams withMapReduce", *Big Data Congress, IEEE*, pp. *387-394*.

[11] AhsanulHaque, Latifur Khan,2013, "MapReduce based Frameworks for Classifying Evolving Data Stream",*13th International Conference on Data Mining Workshops (ICDMW), IEEE*, pp.1113 – 1120.

[12] AhsanulHaque, Brandon Parker, Latifur Khan,and BhavaniThuraisingham, , 2013 "Intelligent MapReduce based Framework for Labeling Instances in Evolving Data Stream",*5th International Conference onCloud Computing Technology and Science (CloudCom), (Volume:2), IEEE*, pp. 299 - 304.

[13] AhsanulHaque, Brandon Parker, Latifur Khan,and BhavaniThuraisingham,2014 "Evolving Big Data Stream Classification withMapReduce",*7th International Conference on Cloud Computing (CLOUD),IEEE* ,pp. 570 – 577.

[14] Ms. Priyanka B.Dongre, Dr. Latesh G. Malik,2014 ,"A Review on Real Time Data Stream Classification and Adapting To Various Concept Drift Scenarios",*International Advance Computing Conference (IACC),IEEE*,pp. 533 - 537.

[15] Domingos, P., and Hulten, G.,2000, "Mining high-speed data streams", *In: Proc. ACMSIGKDD, Boston, MA, USA,ACM Press, New York*, pp. 71–80.

[16] Gehrke, J., Ganti, V., Ramakrishnan, R., and Loh, W. , 1999,"Boat–optimistic decision tree Construction", *In: Proc. ACM SIGMOD, Philadelphia, PA, USA*, pp. 169–180.

[17] Hulten, G., Spencer, L.,and Domingos, P. ,2001, "Mining time-changing data streams", *In:Proc. ACM SIGKDD, San Francisco, CA, USA*, pp. 97–106.

[18] Utgoff, P.E. ,1989,"Incremental induction of decision trees", *Machine Learning 4*, pp.161–186.

[19] Freund, Y.,and Schapire, R.E. ,1996,"Experiments with a new boosting algorithm",*In: Proc. International Conference on Machine Learning (ICML), Bari, Italy*, pp. 148–156.

[20] Scholz, M.,and Klinkenberg., R. ,2005,"An ensemble classifier for drifting concepts", *In: Proc. Second International Workshop on Knowledge Discovery in Data Streams(IWKDDS), Porto, Portugal*, pp. 53–64.

[21] Kolter, J.Z., and Maloof, M.A. ,2005," Using additive expert ensembles to cope with concept drift", *In: Proc. International conference on Machine learning (ICML), Bonn, Germany*, pp. 449–456.

[22] Gao, J., Fan, W., and Han, J. ,2007," On appropriate assumptions to mine data streams", *In: Proc. IEEE International Conference on Data Mining (ICDM), Omaha, NE, USA*, pp. 143–152.

[23] M. Masud, J. Gao, L. Khan et al. ,2008, "A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data", *In Proc. Of Eighth IEEE International Conference on Data Mining, ICDM_'08*,pp. 929 - 934.

[24] P. Zhang, X.Zhu,and L.Guo,2009 ,"Mining data streams with labeled and unlabeled training examples", *In Proc. of Ninth IEEE International*

Conference on Data Mining, ICDM '09,pp.627 - 636.

[25] Fan, W. ,2004,"Systematic data selection to mine concept-drifting data streams", *In: Proc.ACM SIGKDD, Seattle, WA, USA*, pp. 128–137.

[26] Scholz, M., Klinkenberg., R., 2005," An ensemble classifier for drifting concepts", *In: Proc. Second International Workshop on Knowledge Discovery in Data Streams(IWKDDS), Porto, Portugal*, pp. 53–64.