Iterative MapReduce for Feature Selection

Ahlem Kourid
Computer Science Department,
College of NTIC, Constantine University 2,
25000 Constantine,
Algeria

***Abstract*— Scale feature selection is one of the most important fields in big data domain that can resolve problems in real data, such bioinformatics when it is necessary to process huge amount of data, the efficiency of existing feature selection algorithms significantly downgrades, if not totally inapplicable, when data size exceeds hundreds of gigabytes, because most feature selection algorithms are designed for centralized computing architecture. For that distributed computing techniques, such as MPI and MapReduce can be applied to handle very large data. Our algorithm is to scale the existing method for feature selection Kmeans clustering and Signal to Noise Ratio (SNR) using iterative MapReduce. We have used parallel Kmeans on MapReduce for clustering features, and then we have applied iterative MapReduce that implement parallel SNR ranking for each cluster, after we have selected the top ranked features from each cluster, finally top ranked features from all clusters are aggregated with combine function and validated using 10 fold cv with both SVM and KNN classifiers. The efficiency of the proposed method is illustrated through analyzing practical problems.**

Keywords*— *Feature selection; Scale Machine Learning; Big Data; Bioinformatics.

I. INTRODUCTION

With the progress of high technology in several fields that produce an important volume of data such Microarray in bioinformatics [8], deal with high dimensional data becomes a challenge for several tasks in machine learning. Feature Selection is one of the techniques of reduction dimensionality [7] that are effective in removing irrelevant data; increasing learning accuracy therefore becomes very necessary for machine learning tasks.

Scalability can become a problem for even simple and centralized approaches, for that feature selection method based on parallel algorithm will be the mainly choice for dealing with large-scale data. Many parallel algorithms are implemented using different parallelization techniques such as MPI, MapReduce. MapReduce is a programming model for distributed computation, derived from the functional programming concepts, and is proposed by Google for large-scale data processing in a distributed computing environment [4].

Feature selection methods are divided into two main categories: filter method that assess the relevance of features by looking only at the intrinsic properties of the data, in most cases a feature relevance score is calculated and low-scoring features are removed, wrapper methods that evaluate a specific subset of features by training and testing a specific classification model [7]. Recent comparison studies of feature selection method in high-dimensional data have shown that the combination of Kmeans clustering and filter method based SNR (Signal to Noise Ratio) score is graceful for classification problem [1], but the existing method is limited over

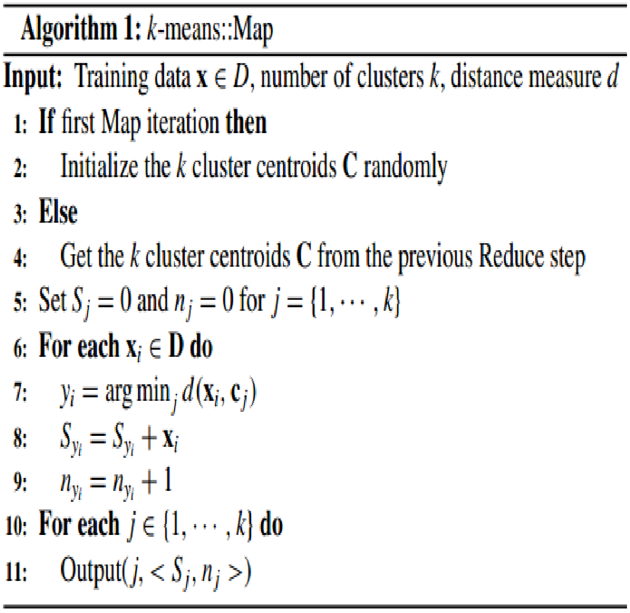
large scale datasets. In order to overcome that problem we present our method that is suitable for very large data, and that have the potential for parallel implementation, based on parallel Kmeans on MapReduce that cluster huge amount of features, so similar features having the same characteristics will be grouped in the same cluster, and on an iterative MapReduce that implement parallel SNR ranking of each cluster. The remaining part of this paper is organized as follows: section 2 introduces the main preliminaries concepts used in this work, section 3 presents the related works, section 4 discusses the algorithm, section 5 deals with the proposed model (Fig1), section 6 explains in details our approach, section7 presents the experiment and results, section 8 discusses the results obtained, and finally section 9 includes a conclusion from the experiment done and presents the future work.

II. PRELIMINARIES

Kmeans on MapReduce

In the MapReduce implementation of k-means, each mapper in the Map phase is assigned a subset of points. For these points, the mapper does the cluster assignment step – it computes yi, the index of the closest centroid for each point xi, and also computes the relevant cluster aggregation statistics:Sj, the sum of all points seen by the mapper assigned to the jth cluster; and nj, the number of points seen by the mapper assigned to the jth cluster. At the end of the Map phase, the cluster index and the corresponding cluster aggregation statistics (sum and counts) are output.

The Map algorithm is shown in Algorithm 1 [9].

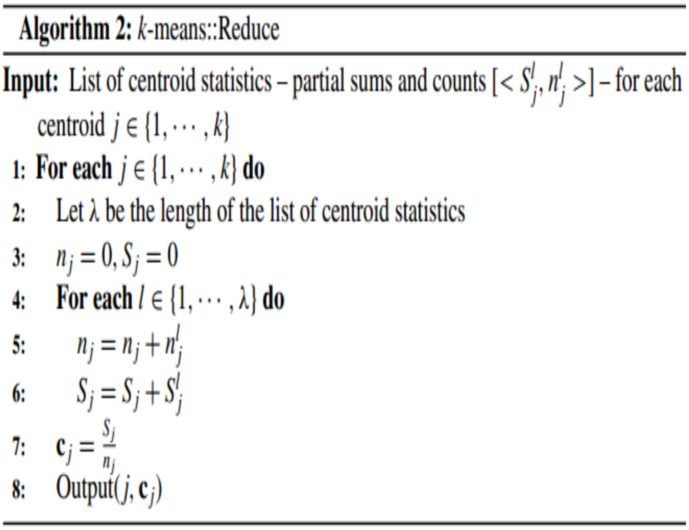


Algorithm 1: *k*-means::Map

```
Input: Training data $\mathbf{x} \in D$, number of clusters $k$, distance measure $d$
If first Map iteration then
    Initialize the $k$ cluster centroids $\mathbf{C}$ randomly
Else
    Get the $k$ cluster centroids $\mathbf{C}$ from the previous Reduce step
Set $S_j = 0$ and $n_j = 0$ for $j = \{1, \cdots, k\}$
For each $\mathbf{x}_i \in \mathbf{D}$ do
    $y_i = \arg\min_j d(\mathbf{x}_i, \mathbf{c}_j)$
    $S_{y_i} = S_{y_i} + \mathbf{x}_i$
    $n_{y_i} = n_{y_i} + 1$
For each $j \in \{1, \cdots, k\}$ do
    Output($j, < S_j, n_j >$)
```

cluster statistics. For each cluster j, the reducer gets a list of cluster statistics [<S^lj,n^lj >], where l is an index over the list – the lth partial sum S^lj in this list represents the sum of some points in cluster j seen by any particular mapper, whereas the lth number n^lj in the list is the count of the number of points in that set. The reducer calculates the average of S^lj to get the updated centroid cj for cluster j.

The Reduce algorithm is shown in Algorithm 2.



Algorithm 2: *k*-means::Reduce

```
Input: List of centroid statistics – partial sums and counts [$< S_j^l, n_j^l >$] – for each
    centroid $j \in \{1, \cdots, k\}$
For each $j \in \{1, \cdots, k\}$ do
    Let $\lambda$ be the length of the list of centroid statistics
    $n_j = 0, S_j = 0$
    For each $l \in \{1, \cdots, \lambda\}$ do
        $n_j = n_j + n_j^l$
        $S_j = S_j + S_j^l$
    $\mathbf{c}_j = \frac{S_j}{n_j}$
    Output($j, \mathbf{c}_j$)
```

SNR (Signal-to-Noise Ratio) is a statistical method and metric that measures effectiveness of feature.
The essential usage in bioinformatics is to locate genes that are differential expressed on Microarray experiment.SNR is defined as follows:

Signal to noise ratio= (µ1-µ2)/(σ1+ σ2)

Where µ1 and µ2 denote the mean expression values for the sample class 1 and class 2 respectively. σ1 and σ2 are the standard deviations for the samples in each class [1]. In our case class 1 and class 2 refer to the target class of each sample (cancer or normal).

- MapReduce

MapReduce is a functional programming model that is well suited to parallel computation. The model is divided into two functions which are map and reduce .In MapReduce; all data are in the form of keys with associated values. For example, in a
program that counts the frequency of occurrences for various words, the key would be a word and the value would be its frequency [11]. MapReduce makes the guarantee that the input to every reducer is sorted by key. The process by which the system performs the sort and transfers the map outputs to the reducers as inputs is known as the shuffle [10].

A MapReduce operation takes place in two main stages. In the first stage, the map function is called once for each input record. At each call, it may produce any number of output records. In the second stage, this intermediate output is sorted and grouped by key, and the reduce function is called once for each key. The reduce function is given all associated values for the key and outputs a new list of values.
The following notation and example are based on the original presentation [13]:

A. Map Function
A map function is defined as a function that takes a single key-value pair and outputs a list of new key-value pairs. The input key may be of a different type than the output keys, and the input value may be of a different type than the output values:

Map :(K1, V1) → list ((K2, V2))

Since the map function only takes a single record, all map operations are independent of each other and fully parallelizable.

B. Reduce Function
A reduce function is a function that reads a key and a corresponding list of values and outputs a new list of values for that key. The input and output values are of the same type.

Reduce :(K2, list (V2)) → list (V2)

A reduce operation may depend on the output from any number of map calls, so no reduce operation can begin until all map operations have completed. However, the reduce operations are independent of each other and may be run in parallel.

MapReduce programs can be run in three modes [12]:

A. Standalone Mode: only run a Java virtual machine, no distributed components. This mode does not use HDFS file system, but use native Linux file system.

B. Pseudo-distributed Mode: start several JVM process on the same machine, each hadoop daemon runs in a separate JVM process, do "pseudo-distributed" operation.

C. Fully-distributed Mode: the real run on multiple machines distributed mode. Standalone mode using the local file system as
well as local MapReducer job runner, distributed mode using HDFS and MapReduce daemons.

- *Classification Techniques Revisited*

SVM: Support Vector Machines (SVMs) are the newest supervised machine learning technique [13]. SVMs revolve around the notion of a “margin”—either side of a hyperplane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error[15].SVM is widely used in the domain of cancer studies, protein identification and especially in Microarray data [17].

K-Nearest Neighbor (KNN): is one of the most straightforward instance-based learning algorithms. KNN is based on the principle that the instances within a dataset will generally exist in close proximity to other instances that have similar properties [14]. If the instances are tagged with a classification label, then the value of the label of an unclassified instance can be determined by observing the class of its nearest neighbors. The KNN locates the k nearest instances to the query instance and determines its class by identifying the single most frequent class label [15].

- Microarray data:

The gene expression microarray technology allows us to measure expressions of thousands of genes simultaneously in a single experiment. This technique presents gene expression of genes under different conditions. A Microarray data set can be represented as an expression table. Where, each row corresponds to a particular gene and each column to a sample. E = {Xij |i=1…...k, j=1,……,n} where Xij ∈ R is the expression level of gene gi sample Si [16].

III. RELATED WORKS

In one hand many works based sample approaches were proposed. [1] combined Kmeans clustering with SNR ranking, the method is improved with SVM and yields a high accuracy .[6][2] used SNR as filtering technique with optimization technique particle swarm optimization PSO in order to hybridized filter and wrapper methods .[3] coupled PNN probabilistic neural network with Signal to Noise Ratio and have achieved better results. [18] proposed a method for selecting informative features (genes) using k-means clustering and SNR ranking, Genetic Programming is used as a classifier. They have compared the experimental results with many feature selection and classifiers among them only KNN using Pearson’s coefficient correlation and information gain as feature selection showed the better result than the proposed approach.
In the other hand we have feature selection related to work in term of parallelization based Mapreduce, those designed for logistic regression, Parallel SFO and parallel grafting [5] that effectively have partitioned the computation first over the records and then over the candidate features. These parallels methods are improved in term of accuracy and time.

IV.ALGORITHM AND DEFINITION OF MAP AND REDUCE FUNCTIONS

- list contains target classes of samples in order.
- record contains values of samples for feature$_i$.
- DFS is a distributed directory system for storage of output and input files of MapReduce.
- ID_feature is an identifier characterizes each feature.
- file_cluster$_i$ contains features of cluster i.

```
 Clustering features with Kmeans on MapReduce.

  For each cluster i do

  DFS.put (file_cluster_i)

  Map function (parallel over features) (key: ID_feature,
  value: record)

          List= [class1, class2, class2……………………]
Iterate over record and list

 compute μ1, μ2
 compute σ1, σ2
 compute SNR

Output (SNR, (ID_feature, record))
```

Reduce Function (key: SNR, value :(ID_feature, record)
Output (SNR, (ID_feature, record))

DFS.delete (file_cluster$_i$)

V. PROPOSED MODEL

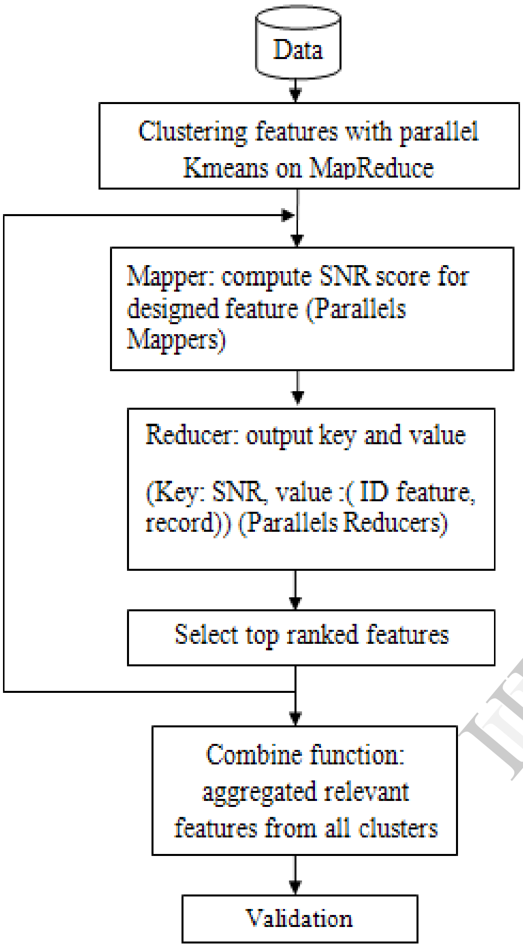

Data
Clustering features with parallel Kmeans on MapReduce
Mapper: compute SNR score for designed feature (Parallels Mappers)
Reducer: output key and value
(Key: SNR, value :(ID feature, record)) (Parallels Reducers)
Select top ranked features
Combine function: aggregated relevant features from all clusters
Validation

Fig1. Model for validation of the proposed method.

VI. APPROACH

Step1: clustering features with parallel Kmeans. As by applying clustering technique we can group similar type of features in same cluster so that best feature from each cluster can be selected.

Step2: mappers read lines (features) and compute SNR score for each feature.

Step3: accord to the paradigm shuffle and sort in MapReduce, the final output files contains ranked SNR values, top ranked features are selected in two cases:

- One reducer: one reducer output one file, and then we select the top ranked features from this file.
- Multiple reducers: multiple reducers output multiple files each file is ranked by SNR value, for that we select the first element from each file, and then find the max one, finally we select the top ranked features from the file that its first element is the max.

Step4: top features ranked from each cluster are aggregated by combine function and validated using classifiers, in our case we have chosen SVM and KNN, go to step 1 for the next cluster.

VII. RESULTS AND EXPERIMENTS

We have used leukemia dataset of cancer Microarray gene expression data, the dataset contains 7,129 genes (features) and 72 samples(47ALL, 25 AML), for our approach we have taken all features of the dataset .The experiment is done on Linux Ubuntu 13.10, using hadoop1.2.1 and mahout 0.9 ,our machine is configurated in pseudo distributed mode. We have applied the cross validation method 10 fold CV to get accuracy. Experiment is done with 5 clusters and 10 clusters. Results are shown on Table I; Table II contain results from literature work. Fig 1; Fig 2 present a comparison of SVM and KNN in the two approaches.

TABLE I: Accuracy of SVM and KNN in our method.

Method	dataset	No of clusters	Genes selected	10 fold CV validation accuracy (%)
(Kmeans+SNR +SVM)	Leukemia 7129 genes	5	5	98,6%
(Kmeans+SNR +KNN)		5	5	98,6%
(Kmeans+SNR +SVM)		10	10	100%
(Kmeans+SNR +KNN)		10	10	95,8%

TABLE II: Accuracy of SVM and KNN in literature [1].

Method	dataset	No of clusters	Genes selected	10 fold CV validation accuracy (%)
(Kmeans+SNR +SVM)	Leukemia 50 genes	5	5	99,3%
(Kmeans+SNR +KNN)		5	5	99,3%
(Kmeans+SNR +SVM)		10	10	94,1%
(Kmeans+SNR +KNN)		10	10	89,3%

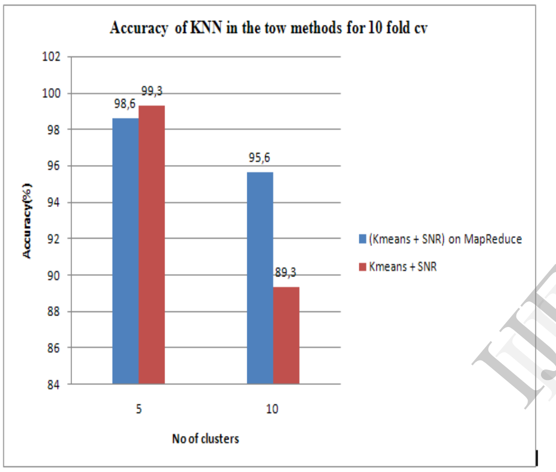


Fig .2: Comparison of accuracy of SVM in the two approaches.

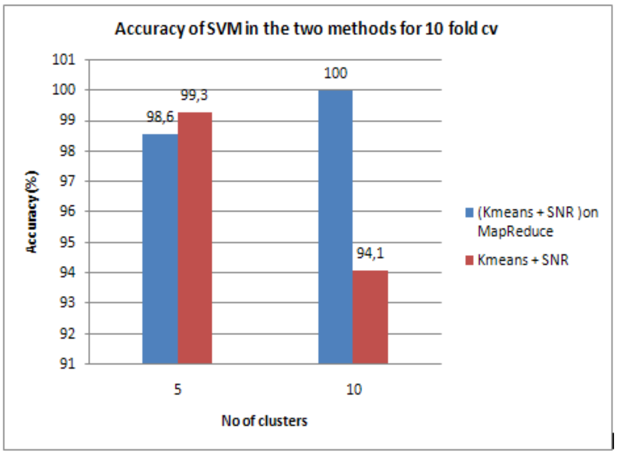


Fig .3: Comparison of accuracy of KNN in the two approaches.

VIII. DISCUSSION RESULTS AND COMPARISON

Results Shows That Our Method Giving An Approximate Values Of Accuracy In Comparison With Centralized Approaches In Literature With The Both Classifiers SVM And KNN, But In Our Approach The Better Accuracy Is Achieved With 10 Clusters With SVM Classifier In The Cross Validation (10 Fold CV). The Methods Presented In Literature Have Shown That The Better Accuracy Is Given With 5 Clusters [1]. We Can Confirm Correctness And Effectiveness Of Our Mapreduce Algorithm. Due To The Recent Comparison Studies And Research, SVM Is Taken As A Most Efficient Classification And Regression Model, And The 10 Fold CV Is The More Accurate Cross Validation For Validation Of Any Research Work, For That Our Results Is Improved. Our Approach Is Graceful In Case Of Problems That Have Big Number Of Features And Few Samples Such Problem Of Diagnostic And Biomarker Discovery In Bioinformatics.

IX. CONCLUSION AND FUTURE WORK

From above results and comparative analysis we can conclude that our method performs well, and gives the same performance as centralized approaches in term of accuracy, for that our method can be applied for large scale datasets and overcomes challenge of feature selection in Big Data, special for biomarker discovery in bioinformatics. Our future work is to improve our approach in term of time using cluster of nodes, and to work on scaling wrapper methods since that wrapper methods gives high performance when be combined with filter methods .

REFERENCES

[1] Debahuti Mishra, Barnali Sahu, “Feature Selection for Cancer Classification : A Signal-to-noise Ratio Approach,” International Journal of Scientific & Engineering Research, Volume 2, Issue 4, April-2011.

[2] Hualong Yu, Guochang Gu, Haibo Liu, Jing Shen, Changming Zhu, “A Novel Discrete Particle Swarm Optimization Algorithm for Microarray Data-based Tumor Marker Gene Selection” , International Conference on Computer science and software Engineering, pp. 1057-1060, 2008.

[3] Chenn-Jung Huang ,Wei-Chen Liao, “A Comparative Study of Feature Selection Methods for Probabilistic Neural Networks in Cancer Classification”, Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’03),Vol 3, pp1082-3409, 2003.

[4] Dean, J., and Ghemawat, S. 2004. MapReduce:Simplified Data Processing on Large Clusters. In : OSDI’04: Sixth Symposium on Operating System Design and Implementation.

[5] Singh, S., Kubica, J., Larsen, S., and Sorokina,. Parallel Large Scale Feature Selection for Logistic Regression. In:SIAM International Conference on Data Mining (SDM), D. 2009.

[6] Barnali Sahu , Debahuti Mishra, “A Novel Feature Selection Algorithm using Particle Swarm Optimization for Cancer Microarray Data” , International Conference on Modeling Optimization and Computing (ICMOC-2012).

[7] Yvan Saeys, Inaki Inza ,Pedro Larranaga, “A review of feature selection techniques in bioinformatics” , Bioinformatics, vol. 23, no. 19, pp. 2507-2517, 2007.

[8] Jay Shendure ,Hanlee Ji, , “Next-generation DNA sequencing” ,Nature Biotechnology vol. 26 no.10 ,2008

[9] R Bekkerman, M Bilenko, J Langford, “Scaling up Machine learning”, Cambridge University Press, 2011.

[10] T White, “hadoop the definitive guide ” , books.google.com., 2012.

[11] Andrew W. McNabb, Christopher K. Monson, and Kevin D. Seppi, , “Parallel PSO Using MapReduce” , Springuer, 2007 .

[12] Gaizhen Yang, “The Application of MapReduce in the Cloud Computing”, IEEE, 2011.

[13] Vapnik, “The Nature of Statistical learning theory”, Springuer, 1995.

[14] Cover, T., Hart, P., “nearest neighbor pattern classification”, IEEE, 1967.

[15] SB Kotsiantis, ID Zaharakis, PE Pintelas, “Supervised Machine Learning: A Review of Classification Techniques” - Artificial Intelligence Review, 2006 – Springer.

[16] Debahuti Mishra, Barnali Sahu, “A Signal-to-noise Classification Model for Identification of Differentially Expressed Genes from Gene Expression Data ” , IEEE, 2011.

[17] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines”, Machine Learning, , Springer, 2002.

[18] Supoj Hengpraprohm, Prabhas Chongstitvatana, “Selecting Informative Genes from Microarray Data for Cancer Classification with GeneticProgramming Classifier Using K-Means Clustering and SNR Ranking”, IEEE, 2007.