DIMENSIONALITY USING OPTIMIZATION ALGORITHM FOR HIGH DIMENSIONAL DATA CLUSTERING

Saranya.S*

Dr.Punithavalli.M**

Abstract:

This paper present an efficient approach to a feature selection problem based on genetic algorithm for high dimensional data clustering. We proposed a new algorithm uses an inconsistency rate to evaluate the fitness of individuals in population which results in a significant reduction in terms of computational time needed to reach a subset of relevant features and finally the relevant features are clustered by clustering algorithm k-means algorithm. The performances of efficient approach of features selection are evaluated on several high dimensional dataset. The experiment result shows that the proposed algorithm is faster process than the existing algorithm for feature selection and increasing the predictive accuracy and the dimensionality is reduced

Keyword: feature selection, Genetic algorithm, data clustering, k-means algorithm.

* Research Scholar, Bharathiyar University, Coimbatore, India.

** Director, Sri Ramakrishna College of Engineering, Coimbatore, India.

I. INTRODUCTION

Clustering is an unsupervised data mining process, which creates the groups (clusters) of objects where objects having similar property belong to one group (cluster) and the objects having dissimilar property belong to another cluster. Conventional clustering methods, such as partitioning algorithms and hierarchical algorithms use prior knowledge of the number of clusters. Partitioning algorithms start with the random selection of *k* objects, from the data set, as initial clusters centers and assign other objects to the nearest cluster center. Hierarchical clustering algorithms create a tree structure, known as dendrogram of the objects in the data set.

All such traditional clustering algorithms create clusters based on the similarity or distance, hence fail to work on high dimensional data set as all objects become equidistant in high dimensional data set; there is no difference between closest and farthest data objects. As partitioning clustering algorithms choose the initial clusters centers randomly, they suffer with the problem of local optimum clusters; in different runs they produce different results. Dimensionality reduction is a major issue in the supervised as well as unsupervised data mining techniques. It reduces the size of the representation of the data objects which also results in less computational cost in further steps. However, a care should be taken that the reduced data set truly represents the original data set, i.e., there should be minimal descriptive inaccuracy. Dimensionality reduction techniques may be categorized into two distinct groups: feature transformation (FT) and feature selection (FS). Feature selection is a pre-processing step that uses ranking or weighting strategy for removing the irrelevant or noisy dimension of the data set for both supervised and unsupervised data mining task. For classification, it selects the dimensions which give the highest accuracy for a class, whereas in clustering, it chooses the dimensions which produce better clusters. There are two well-known methods for unsupervised feature selection: filter methods and wrapper methods.

John G. H., Kohavi R. and Pfleger K (1994) “Irrelevant features and the subset selection Problem” has proposed a solution to finding the subset of features that allows a supervised Induction algorithm to induce small high accuracy concept. They present a definition for irrelevance and two degree of relevance. This definition has improve the previous subset selection algorithm and help to subset of features that should be a sought. The feature selection should depend not only on features and target concept, but also on the induction algorithm. The description method for feature subset selection using cross validation that is applicable to any induction algorithm.

M. Richeldi, P. L. Lanzi (1996) we propose an alternative approach to GA-based feature selection, in which a preprocessing step, named *Data Reduction* (DR), is applied to inspect the underlying structure of the data and initialize the genetic search algorithm. The data reduction step is aimed at reducing the dimensionality of the data and is independent of the knowledge discovery algorithm. An iterative, statistical method is applied to explore direct (first-order) and indirect (higher-order) dependencies between data and to partition the set of observed features into a small number of clusters (factors) such that those features in a given cluster can be regarded as measures of the same underlying construct. A genetic algorithm has been designed that explores the factors obtained in the data reduction step and determines the most informative subset of features. The fitness of the genetic algorithms is the performance of the induction algorithm employed in the knowledge extraction step.

Huan Liu , Rudy Setiono “A Probabilistic Approach to Feature Selection - A Filter Solution“ has proposed a probabilistic approach. The theoretic analysis and the experimental study show that the proposed approach is simple to implement and guaranteed to find the optimal if resources permit. It is also fast in obtaining results and effective in selecting features that improve the performance of a learning algorithm. An on-site application involving huge datasets has been conducted independently. It proves the effectiveness and scalability of the proposed algorithm.

Lydia Boudjeloud and Fran¸cois Poulet “Attribute Selection for High Dimensional Data Clustering” One possible solution is to use only a subset of the whole set of dimensions. But the number of possible dimension subsets is too large to be fully parsed. We use a heuristic search for optimal attribute subset selection. For this purpose we use the best cluster validity index to first select the most appropriate cluster number and then to evaluate the clustering performed on the attribute subset.

II. EXISTING APPROACH:

The Genetic algorithm is an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500.the genetic algorithm deals with isolating the very few relevant features from the large set. This is not exactly the classical feature selection problem known in Data mining as, for example around 50% of features are selected. Here, we have the idea that less than 5% of the features have to be selected. But this problem is close from the classical feature selection problem and we will use a genetic algorithm as we saw they are well adapted for problems with a large number of features. We present here the main characteristics and adaptations we made to deal with this particular feature selection problem. Our genetic algorithm has different phases. It proceeds for a fixed number of generations. A chromosome, here, is a string of bits whose size corresponds to the number of features. A 0 or 1, at position i, indicates whether the feature i, is selected (1) or not (0).

The genetic operators: These operators allow GAs to explore the search space. However, operators typically have destructive as well as constructive effects. They must be adapted to the problem.

Crossover: We use a Subset Size-Oriented Common Feature Crossover Operator (SSOCF) which keeps useful informative blocks and produces offspring's which have the same distribution than the parents. Offspring's are kept, only if they fit better than the least good individual of the population. Features shared by the 2 parents are kept by offspring's and the non shared features are inherited by offspring's corresponding to the i^{th} parent with the probability (n_i-n_c/n_u) where n_i is the number of selected features of the i^{th} parent, n_c is the number of commonly selected features across both mating partners and nu is the number of non-shared selected features.

Mutation**:** The mutation is an operator which allows diversity. During the mutation stage, a chromosome has a probability p_{mut} to mutate. If a chromosome is selected to mutate, we choose randomly a number n of bits to be flipped then n bits are chosen randomly and flipped. In order to create a large diversity, we set p_{mut} around 10%.

Selection: We implement a probabilistic binary tournament selection. Tournament selection holds n tournaments to choose n individuals. Each tournament consists of sampling 2 elements of the population and choosing the best one with a probability p

Specific adaptations and mechanisms**:** The chromosomal distance (A distance adapted to the problem): The biologist experts indicate that a gene is correlated with its neighbours situated on the same chromosome at a distance smaller than σ equals to 20 CMorgan (a measure unit). So in order to compare two individuals, we create a specific distance which is a kind of bit to bit distance where not a single bit i is considered but the whole window (i- s, i+s) of the two individuals are compared. If one and only one individual has a selected feature in this window, the distance is increased by one.

The fitness function**:** The fitness function we developed refers to the support notion, for an association, which, in data mining, denotes the number of times an association is met over the number of times at least one of the members of the association is met. The function is composed of two parts. The first one favours for a small support a small number of selected features because biologists have in mind that associations will be composed of few features and if an association has a bad support, it is better to consider less features (to have opportunity to increase the Support) The second part, the most important (multiplied by 2), favours for a large support a large Number of features because if an association has a good support, it is generally composed of few features and then we must try to add other features in order to have a more complete association. What is expected is to favours good associations (in term of support) with as much as features as possible. This expression may be simplified, but we let it in this form in order to identify the two terms.

F = (1-S)..(T/10-10.SF)̃ T).2.(S..(T/10-10.SF)/T

Where:

S = □ (□ □ □ □ □ …)/(□ □ □ □ □ …)□ where, □ … are the Selected features

T = Total number of features,

SF = Number of selected significant features

The working process of genetic algorithm

Step 1: Represent the problem variable domain as a chromosome of fixed length; choose the size of the chromosome population N, the crossover probability Pc and the mutation probability P_m.

Step 2: Define a fitness function to measure the performance of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

Step 3: Randomly generate an initial population of size N: $x_1, x_2, \ldots, x_N$

Step 4: Calculate the fitness of each individual chromosome: $f(x_1), f(x_2), \ldots, f(x_N)$

Step 5: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. High fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

Step 6: Create a pair of offspring chromosomes by applying the genetic operators.

Step 7: Place the created offspring chromosomes in the new population.

Step 8: Repeat Step 5 until the new population size equals that of the initial population, N.

Step 9: Replace the initial (parent) chromosome population with the new (offspring) Population.

Step 10: Go to Step 4, and repeat the process until the termination criterion is satisfied.

III. PROPOSED APPROACH:

The genetic algorithm proposed for feature selection purposes maintain the above representation and the standard genetic operators. Instead the fitness of individuals is computed using the inconsistency rate. The inconsistency rate specifies to what extent the reduced data still represent the original dataset and can be considered a measure of how much inconsistent the data become when only a subset of attributes is considered. Consider now Figure 1 where two items of a dataset with four attributes {$att_{l.}$ . . att_4} and one class, class, with values { class attr cl_O, class attr cl_1) is shown.

Items	Att_1	Att_2	Att_3	Att_4	Class atrri
…					…
$Item_i$	0	1	2	4	Cl_0
…					…
$Item_j$	0	1	3	4	Cl_1
…					…

The two items have different class values but differs only in attribute att_3 so that if the feature subset {att_l, att_2, att_4) is considered we get an inconsistency in the data. In Figure **2** where only the subset {att_l, att_2, att_4) is considered the two items are equal with respect to attribute values but differs for class attribute values, there is an example that has been classified with two different labels: this is inconsistent.

Fig. **2.** The **two** items, item, and item, in the previous dataset when **only** the subset **of** features

{ att_l, att_2, att_4} **is** considered.

Inconsistency is introduced in the data when the number of attributes is reduced; the rate measures how much inconsistency is introduced when only a certain feature subset is considered.

The rate is computed as follows:

Step 1: Two items of the given dataset are considered inconsistent if they match except for they class labels with respect to the subset of features considered;

Step 2: for all matching instances the inconsistency count is the number *n* of instances minus the largest number of instances of the most frequent class label; for example if there are two class label c_1 and c_2 with respectively

n_l and n_2 instances ($n_1 + n_2 = n$) then the inconsistency count is equal to (TI - max(n_1,n_2)).

Step 3: The inconsistency rate is computed as the quotient of the sum of all the inconsistency counts divided by the total number of instances.

In existing algorithm the dataset usually cross-validation is employed to avoid over-fitting. These problem has overcome is the proposed algorithm by inconsistency rate is a simple statistics and doesn't require cross-validation. Nevertheless the rate is only an approximated measure of the information loss when a subset of features is considered thus may be non informative in some cases.

The clustering phase:

***Use of k-means algorithm*:** The k-means algorithm is an iterative procedure for clustering which requires an initial classification of the data. The k-means algorithm proceeds as follows: it computes the center of each cluster, and then computes new partitions by assigning every object to the cluster whose center is the closest (in term of the Hamming distance) to that object. This cycle is repeated during a given number of iterations or until the assignment has not changed during one iteration. Since the number of features is now very small, we implement a classical k-means algorithm widely used in clustering and to initialize the procedure we randomly select initial centers.

IV EXPERIMENTAL AND RESULTS:

We evaluated the proposed new algorithm data partitioning based k-means on glass, segment dataset taken from the UCL repository of machine learning databases, is used for testing the accuracy of the cluster using proposed new algorithm.

The performance of the proposed algorithm is evaluated using the following parameters,

- Test the effectiveness to evaluate fitness
- Number of features in the original data and in the reduced data

***Glass dataset*:**

The study of classification of types of glass was motivated by criminological investigation. Number of Attributes: 10 (including an Id#) plus the class attribute

The attributes are,

- Id number: 1 to 214
- RI: refractive index
- Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
- Mg: Magnesium
- Al: Aluminum
- Si: Silicon
- K: Potassium
- Ca: Calcium

- Ba: Barium
- Fe: Iron

Segment dataset:

The instances were drawn randomly from a database of 7 outdoor images. The images were hand segmented to create a classification for every pixel.

The Attributes are:

- Region-centroid-col: the column of the center pixel of the region.
- Region-centroid-row: the row of the center pixel of the region.
- Region-pixel-count: the number of pixels in a region = 9.
- Short-line-density-5: the results of a line extraction algorithm that counts how many lines of length 5 (any orientation) with low contrast, less than or equal to 5, go through the region.
- Short-line-density-2: same as short-line-density-5 but counts lines of high contrast, greater than 5.
- Vedge-mean: measure the contrast of horizontally adjacent pixels in the region. There are mean and standard deviation are given. This attribute is used as a vertical edge detector.
- vegde-sd
- Hedge-mean: measures the contrast of vertically adjacent pixels. Used for horizontal line detection.
- hedge-sd.
- intensity-mean: the average over the region of (R + G + B)/3
- Rawred-mean: the average over the region of the R value.
- Raw blue-mean: the average over the region of the B value.
- Raw green-mean: the average over the region of the G value.

- exred-mean: measure the excess red: (2R - (G + B))
- exblue-mean: measure the excess blue: (2B - (G + R))
- exgreen-mean: measure the excess green: (2G - (R + B))
- Value-mean: 3-d nonlinear transformation of RGB. (Algorithm can be found in Foley and Van Dam, Fundamentals of Interactive Computer Graphics)
- saturatoin-mean
- hue-mean

***Comparison with the results*:**

Table1: shows the predictive accuracy of glass dataset on the original features and on the reduced data obtained with the proposed algorithm.

	Existing algorithm	Proposed new algorithm
No of clusters(k)	2	2
Accuracy (%)	60.3%	65.1%
No of features	9	5

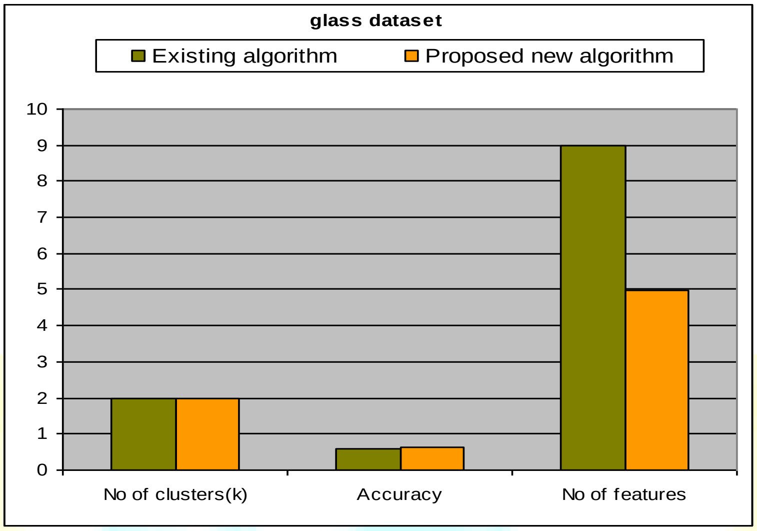

Figure 1: resulted for glass dataset

The Figure 1 represents the resulted of the predictive accuracy and also shown the number of features in the original data and in reduced data according to corresponding cluster.

Table2: shows the predictive accuracy of segment dataset on original features and on the reduced data obtained with the proposed algorithm. It also reported the number of feature in the original data and in the reduced data according to the number of clusters

	Existing algorithm	Proposed new algorithm
No of clusters(k)	2	2
Accuracy(%)	94.7%	95.1%
No of features	19	8

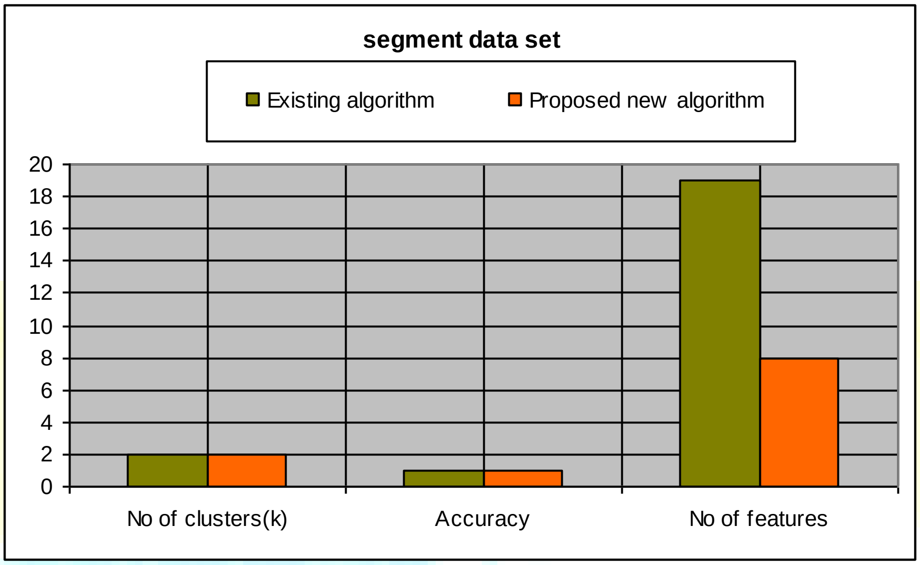

Figure 2: resulted for segment dataset

The Figure 2 represents the resulted of the predictive accuracy and also shown the number of features in the original data and in reduced data according to corresponding cluster.

V.CONCLUSION

The large number of dimensions of the dataset is one of the major difficulties encountered in determining. Most of the existing algorithm looses some of the efficiency in high dimensional dataset. This paper presents an efficient solution to feature selection problem based on genetic algorithm paradigm that fits the feedback model. The proposed new algorithm uses an inconsistency rate to evaluate the fitness of individuals in population which results in a significant reduction in terms of computational time needed to reach a subset of relevant features and finally the relevant features are clustered by clustering algorithm k-means algorithm. The experimental results show that the proposed algorithm performs better than existing algorithm and there is a large gap between the indexes computed with whole data set and the indexes calculated with features subset and dimensionality is reduced.

REFERENCES

[1] Dietterich T. G.: Statistical Tests for Comparing Supervised Classification Learning Algorithms. Tech. Report. Department of Computer Science. Oregon State University. (1996)

[2] John G. H., Kohavi R. and Pfleger K.: Irrelevant Features and the Subset Selection Problem. Proc. of the 11th International Conference on Machine Learning. (1994)

[3] Richeldi M., Lanzi P. L.: Improving Genetic Based Feature Selection by Reducing Data Dimensionality Proc. of the ICML Workshop on Evolutionary Computation. Bari (1996)

[4] Liu H., Setiono R.: A Probabilistic Approach to Feature Selection: A Filter Solution Proc. of the 13th International Conference on Machine Learning. Bari, Italy. (1996)

[5] pier Luca Lanzi: fast feature selection with genetic algorithms: A filter approach: Dipartimento di elettronica e informazione politecnico di milano IEEE (1997)

[6] V.N. Rajavarman and S.P. Rajagopalan: Feature Selection in Data-Mining for Genetics Using Genetic Algorithm (2007)

[7] Merz C and Murphy P, UCI Repository of Machine Learning Databases, Available:

ftp://ftp.ics.uci.edu/pub/machine-learningdatabases