IJDCST @June-July-2015, Issue- V-3, I-5, SW-24
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Advanced Chameleon Algorithm for feature selection

Adavi Sridurga, S.Suresh Babu
[1]**Dept of CSE, Dept. Of CSE ,Sri Mittapalli Clg Of Engineering**
[2]**Assistant Professor, Dept. Of CSE ,Sri Mittapalli Clg Of Engineering**

Abstract: Feature selection is one of the most important methods which involve getting the most useful features that are related to the entire dataset.
In this paper, Fast clustering based feature selection algorithm (FAST) and adopted with advanced chameleon is implemented. Fast is tree-based algorithm and advanced chameleon is graph-based algorithm. Each cluster consists of related items with features. Features in different clusters are relatively independent; the clustering-based strategy of Chameleon has a high probability of producing a subset of useful and independent features. In this paper, to improve the performance of feature extraction proposed system integrated with K-Nearest neighbor graph clustering method. The results will show the efficiency and effectiveness of proposed work.

Key Words: Index Terms - Feature subset selection, filter method, feature clustering, graph-based clustering, Tree-based Clustering.

I. INTRODUCTION

Feature selection, otherwise called variable selection, attribute selection or variable subset selection, is the process of selecting a subset of applicable components for utilization in model development. The focal presumption when utilizing an element determination strategy is that the information contains selected features, and irrelevant features.

Repetitive components are those which give no more data than the as of now chose elements, and superfluous elements give no valuable data in any connection. Feature extraction systems are a subset ofthe more broad field of feature extraction. Feature extraction makes new components from elements of the first elements, though include choice returns a highlights' subset. Feature extraction procedures are frequently utilized as a part of spaces where there are numerous components and nearly few specimens (or information focuses).

A feature selection algorithm can be seen as the mix of a quest strategy for proposing new element subsets, alongside an assessment measure which scores the distinctive element subsets. The simplest algorithm is to test every conceivable subset of elements discovering the particular case that minimizes the lapse rate. This is a comprehensive inquiry of the space, and is computationally obstinate for everything except the littlest of capabilities. The decision of assessment metric vigorously impacts the algorithm.

The majority of real-world classification problems require supervised learning where the underlying class probabilities are obscure, and every occurrence is connected with a class mark. In certifiable circumstances. Consequently, numerous applicant elements are acquainted with better speak to the space. Sadly a large number of these are either in part or totally superfluous/excess to the objective idea. An important component is neither superfluous nor

repetitive to the objective idea; an insignificant element does not influence the objective idea at all, and an excess element does not add anything new to the objective idea. In numerous applications, the span of a dataset is large to the point that learning may not fill in too before evacuating these undesirable elements.

This helps in showing signs of improvement understanding into the basic idea of a genuine characterization issue. Highlight determination techniques attempt to pick a subset of elements that are pertinent to the objective idea. Highlight choice is characterized by numerous creators by taking a gander at it from different edges. Be that as it may, of course, a number of those are comparative in instinct and/or substance. So in this paper we talk about the element subset calculations FAST and Chameleon order calculations.

And following sections are 2. Existing system of FAST algorithms 3. Proposed System of Chameleon classification 4.Experment Results 5.Peformances 6.Conclusion.

II. EXISTING SYSTEM

Existing system describe feature selection sub set procedure era in late application administration with determined result amassing. Taking into account the procedure displayed some time recently, we build up a algorithm, named FCBF (Fast Correlation-Based Filter). As in Figure, given an information set with N components and a class C , the algorithm discovers an arrangement of prevalent elements S best for the class idea. It comprises of two noteworthy parts. In the first part (line 2-), it figures the SU esteem for every element, chooses significant elements into S0 rundown taking into account the predefined limit ± , and orders them in plunging request as indicated by their SU values. In the second part (line 8-20), it further procedures the requested rundown S0 rundown to evacuate excess elements and just keeps transcendent ones among all the chose important components. As indicated by Heuristic 1, a component Fp that has as of now been resolved to be a prevalent element can simply be utilized to sift through different elements that are positioned lower than Fp and have Fp as one of its excess companions. The cycle begins from the Ørst component (Heuristic 3) in S 0 rundown (line 8) and proceeds as takes after. For all the remaining components (from the one privilege by F p to the last one in S 0 rundown), if F p happens to be an excess associate to an element Fq will be expelled from S0 list (Heuris-tic 2).

After one round of sifting components in light of Fp , the algorithm will take the as of now remaining element right beside Fp as the new reference (line 19) to rehash the separating procedure. The calculation stops until there is no more component to be expelled from S0 list.

Highlight subset determination can be seen as the procedure of recognizing and uprooting whatever number insignificant and repetitive components as would be prudent. This is on the grounds that immaterial components don't add to the prescient precision and excess elements don't redound to showing signs of improvement indicator for that they give generally data which is as of now present in different feature(s). Of the numerous component subset determination calculations, some can successfully wipe out unessential elements yet neglect to handle repetitive elements yet some of

others can dispense with the superfluous while dealing with the redundant features.

Algorithm 1: FAST

inputs: $D(F_1, F_2, ..., F_m, C)$ - the given data set
θ - the *T-Relevance* threshold.
output: S - selected feature subset .
//=== Part 1 : Irrelevant Feature Removal ===
for $i = 1$ *to* m **do**
  T-Relevance = SU (F_i, C)
  if *T-Relevance* $> \theta$ **then**
    $S = S \cup \{F_i\}$;
//=== Part 2 : Minimum Spanning Tree Construction ===
G = NULL; *//G is a complete graph*
for *each pair of features* $\{F'_i, F'_j\} \subset S$ **do**
  F-Correlation = SU (F'_i, F'_j)
  Add F'_i *and/or* F'_j *to G with F-Correlation as the weight of the corresponding edge;*
minSpanTree = Prim (G); *//Using Prim Algorithm to generate the minimum spanning tree*
//=== Part 3 : Tree Partition and Representative Feature Selection ===
Forest = minSpanTree
for *each edge* $E_{ij} \in$ *Forest* **do**
  if $\mathrm{SU}(F'_i, F'_j) < \mathrm{SU}(F'_i, C) \wedge \mathrm{SU}(F'_i, F'_j) < \mathrm{SU}(F'_j, C)$ **then**
    Forest = Forest $- E_{ij}$
$S = \phi$
for *each tree* $T_i \in$ *Forest* **do**
  $F_R^j = \mathrm{argmax}_{F'_k \in T_i} \mathrm{SU}(F'_k, C)$
  $S = S \cup \{F_R^j\}$;
return S

Figure 2: FAST Clustering algorithm specification with data set extraction.

Dataset for FAST Algorithm prepared from cars company data prepare the data sets and the store the previous data by using this previous data to give the input to FAST algorithm for the purpose of the to remove un relevant data and make decision for relevant data.

width	height	engine-size	fuelsystem	stroke	horse-power	peak-rpm	price
66.2	54.3	109	mpfi	3.4	102	5500	13950
66.4	54.3	136	mpfi	3.4	115	5500	17450
66.3	53.1	136	mpfi	3.4	110	5500	15250
71.4	55.7	136	mpfi	3.4	110	5500	17710
71.4	55.7	136	mpfi	3.4	110	5500	18920
71.4	55.9	131	mpfi	3.4	140	5500	23875
64.8	54.3	108	mpfi	2.8	101	5800	16430
64.8	54.3	108	mpfi	2.8	101	5800	16925
64.8	54.3	164	mpfi	3.19	121	4250	20970
64.8	54.3	164	mpfi	3.19	121	4250	21105
66.9	55.7	164	mpfi	3.19	121	4250	24565
66.9	55.7	209	mpfi	3.39	182	5400	30760
67.9	53.7	209	mpfi	3.39	182	5400	41315
70.9	56.3	209	mpfi	3.39	182	5400	36880
62.5	54.1	110	1bbl	3.58	86	5800	10295
65.2	54.1	110	mpfi	3.58	101	5800	12945
66	51	110	2bbl	3.58	100	5500	10945
65.6	52.8	230	mpfi	4.17	176	4750	32250
65.6	52.8	230	mpfi	4.17	176	4750	35550
70.6	47.8	326	mpfi	3.76	262	5000	36000

Figure 2.1: Car company dataset

III. PROPOSED APPROACH

In this paper, Advanced CHAMELEON, a new clustering algorithm that overcomes the limitations of existing agglomerative hierarchical clustering algorithms discussed in Section 3. Figure 6 provides an overview of the overall approach used by Advanced CHAMELEON to find the clusters in a data

Advanced CHAMELEON operates on a sparse graph in which nodes represent data items, and weighted edges represent similarities among the data items. This sparse graph representation of the data set allows Advanced CHAMELEON to scale to large data sets and to operate successfully on data sets that are available only in similarity space [GRG+99] and not in metric spaces [GRG+99]. Advanced CHAMELEON finds the clusters in the data set by using a two phase algorithm. During the first phase, CHAMELEON uses a graph partitioning algorithm to cluster the data items into a large number of relatively small sub-clusters. During the second phase, it uses an agglomerative hierarchical clustering algorithm to find the genuine clusters by repeatedly combining together these sub-clusters.

The key feature of CHAMELEON's agglomerative hierarchical clustering algorithm is that it determines the pair of most similar sub-clusters by taking into account both the inter-connectivity as well as the closeness of the clusters; and thus it overcomes the limitations discussed .that result from using only one of them. Furthermore, CHAMELEON uses a novel approach to model the degree of inter-connectivity and closeness between each pair of clusters that takes

into account the internal characteristics of the clusters themselves.

Algorithm:

```
1void heapsort (array_of_nos, int n)
2 {
3 buildHp(array_of_nos,n);
4 shrinkHp(array_of_nos,n);
5 }
6 void buildHp (array_of_nos,n)
7 {
8loop the three steps bellow till all nodes are checked;
9 chld = I - 1;
10 prnt = (chld - 1) / 2;
11 make maximum of children as parents
12 }
13 void shinkhp(array_of_nos,n)
14 {
15 //here each thread is assigned to a particular parent node
16 prnt=0;
//start from root
17 compare left and right child and make maximum as parent;
18 take the max heap from each thread thereby getting each parent node ;
19 i.e the nodes having right and left child ;
20 knowing the position of these set of nodes construct others;
21 }
22 levelorder()
23 {
24 traverse heap in level - order by dividing these levels to threads;
25 connect only siblings to form a graph ;
26 }
```

Parallel K - NN clustering with heap sorting algorithm Pseudo code of parallel merging algorithms for final clusters

1 RI – Relative inter connectivity 2 RC Relative Closeness 3 α - user defined parameter 4 β – RI x RC 5 th – threshold value to take merging decision

6 n be number of clusters to be merge

7 Algorithm : 8

for i=0 ... n // i and j are used for clusters

a . for j=i+1 ... n I . Assign task to work pool merge (i,j); ii .

End for // iteration

IV. EXPERIMENTAL ANALAYSIS

The quality of data partitions generated by this technique is assessed against those created by different categorical data clustering algorithms and cluster ensemble techniques. By knowing the result we cannot estimates the performances of the algorithm that why we apply the data sets on both algorithm get the results and compares the algorithm on the basis on time complexity and space complexity.

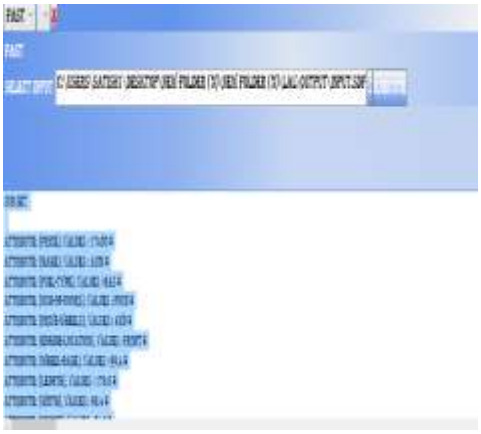

Figure 4.1:FAST Clustering algorithm results

SUB SET

ATTRIBUTE: (PRICE), VALUES : 17450
ATTRIBUTE: (MAKE), VALUES : AUDI
ATTRIBUTE: (FUEL-TYPE), VALUES : GAS
ATTRIBUTE: (NUM-OF-DOORS), VALUES : FOUR
ATTRIBUTE: (DRIVE-WHEELS), VALUES : 4WD
ATTRIBUTE: (ENGINE-LOCATION), VALUES : FRONT
ATTRIBUTE: (WHEEL-BASE), VALUES : 99.4
ATTRIBUTE: (LENGTH), VALUES : 176.6
ATTRIBUTE: (WIDTH), VALUES : 66.4

Figure 4.2:FAST Clustering algorithm results remove the un relevant data sets

Figure 4.3: **Advanced CHAMELEON**s Hierarchical model algorithm results

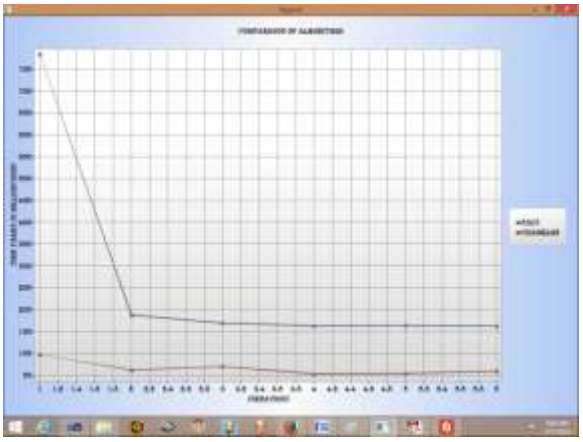

Figure 4.4:Time Taken for **Advanced CHAMELEON** and FAST algorithms.

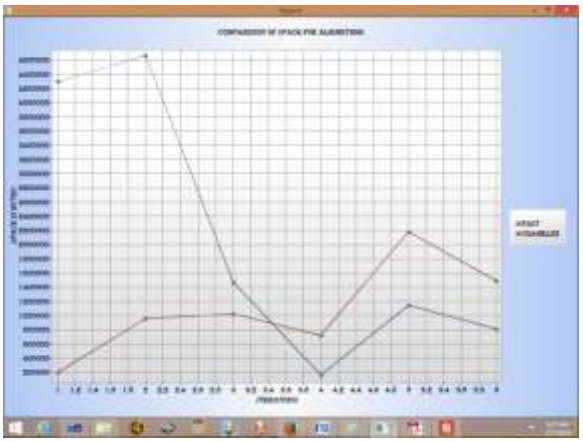

Figure 4.5:Space Taken for **Advanced CHAMELEON** and FAST algorithms.

V. CONCLUSION

We utilize the used dataset of car company set by the clients who are basically utilized the ideal decisions are clients . what's more, make it as an information sets and connected on both FAST clustering algorithm and CHAMELEON various leveled model algorithm and both are utilized for to uproot the unessential data and make it as irrelevant information use get ready for the clustering data sets . In this paper we are analyze the execution by utilizing of time many-sided quality the CHAMELEON calculation give the best results similar FAST clustering algorithms. Furthermore, FAST Clustering algorithm give the best results for space occupation relative to the CHAMELEON algorithm. So at long last on the premise of time many-sided quality CHAMELEON algorithm propelled algorithm for to uproot superfluous information sets and demonstrated by the results.

VI. REFERENCES

[1] Qinbao Song, Jingjie Ni and Guangtao Wang "A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data".

[2] Almuallim H. and Dietterich T.G., Learning boolean concepts in the presence of many irrelevant features, Artificial Intelligence, 69(1-2), pp 279- 305, 1994.

[3] Arauzo-Azofra A., Benitez J.M. and Castro J.L., A feature set measure based on relief, In Proceedings of the fifth international conference on Recent Advances in Soft Computing, pp 104-109, 2004.

[4] Baker L.D. and McCallum A.K., Distributional clustering of words for text classification, In Proceedings of the 21st Annual international ACM SIGIR Conference on Research and Development in information Retrieval, pp 96- 103, 1998.

[5] Battiti R., Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks, 5(4), pp 537-550, 1994.

[6] Bell D.A. and Wang, H., A formalism for relevance and its application in feature subset

selection, Machine Learning, 41(2), pp 175-195, 2000.

[7] Biesiada J. and Duch W., Features election for high-dimensionaldatała Pearson redundancy based filter, AdvancesinSoftComputing, 45, pp 242C249, 2008.

[8] Butterworth R., Piatetsky-Shapiro G. and Simovici D.A., On Feature Se- lection through Clustering, In Proceedings of the Fifth IEEE international Conference on Data Mining, pp 581-584, 2005.

[9] Cardie, C., Using decision trees to improve case-based learning, In Pro- ceedings of Tenth International Conference on Machine Learning, pp 25-32, 1993.

[10] Chanda P., Cho Y., Zhang A. and Ramanathan M., Mining of Attribute Interactions Using Information Theoretic Metrics, In Proceedings of IEEE international Conference on Data Mining Workshops, pp 350-355, 2009