Implementation of DB-Scan in Multi-Type Feature CoSelection for Clustering

K.Parimala [1], Dr. V.PalaniSamy[2]

**[1] Asst Professor, MCA Department, NMS S.Vellaichamy Nadar College,
Madurai-625019, TamilNadu, INDIA**

**[2] Professor & Head of the Deparment, Department of Computer Science & Engineering,
Alagappa University, Karaikudi, TamilNadu, INDIA**

Abstract

Feature Selection is a preprocessing technique in supervised learning for improving predictive accuracy while reducing dimension in clustering and categorization. Multitype Feature Coselection for Clustering (MFCC) with hard k-means is the algorithm which uses intermediate results in one type of feature space enhancing feature selection in other spaces, better feature set is co-selected by heterogeneous features to produce better cluster in each space. Db-Scan is a density-based clustering algorithm finding a number of clusters starting from the estimated density distribution of corresponding nodes. It is one of the most common clustering algorithms and also most cited in scientific literature, as a generalization of DBSCAN to multiple ranges, effectively replacing the εparameter with a maximum search radius.This paper presents the empirical results of the MFCC algorithm with Db-scan and also gives the comparison results of MFCC with hard k-means and DB-Scan. DB-Scan clustering is proposed for getting the quality clustering against the outliers and time criteria is less than any other clustering in high density data set.

Keywords: *Feature Selection, MFCC, Db-Scan.*

1. Introduction

Information or knowledge can be conceptualized as data. It reflects in the data norm, the size and dimensions have improved high and more. The feature selection plays a vital role in machine learning, data mining, information retrieval, etc. the goal of feature selection is to identify those features relevant to achieve a predefined task. Many researchers have been to find how to search feature subset space and evaluate them.

In supervised methods [1], the correlation of each feature with the class label is computed by distance, information dependence or consistency measures [2]. In unsupervised method the feature selection does not need the class of information such as document frequency and term strength [3]. The newly proposed methods namely Entropy based feature ranking method (En) proposed by Dash and Liu [4] in which feature importance is measured by the contribution to an entropy index based on the data similarity; the individual 'feature saliency' is estimated and an Expectation Maximization (EM) algorithm using Minimum message length is derived to select the feature subset and the number of clusters [5].

While the methods above are not directly targeted to clustering text documents, [6] proposes two other feature selection methods for text clustering. One is Term Contribution (TC) which ranks the feature by it overall contribution to the document similarity in the data set. The other is Iterative feature selection (IF), which utilizes some successful feature selection methods such as Information Gain (IG) and CHI-Square (χ^2) text to iteratively select features and performs text clustering at the same time.

[7] Combines information about document contents and hyper link structures to cluster documents. The hypertext documents in a certain information space were clustered into a hierarchical form based on contents as well as link structure of each hyper text documents.

From the ideas of [8] & [9] co-training algorithms learn through classifiers over each of the feature set and combine their predictions to decrease classification error. Cot raining algorithm can learn from unlabelled data starting from a weak predictor.

Clustering helps users, tackle the information overload problem in several ways: explore the contents of a document collection; group duplicate and near duplicate documents. Unsupervised method can hardly achieve a good performance when evaluated using labeled data.

Data fusion [10] is well suited to-problems involving massive amounts of data where each subsystem may not have entire data set, problems with many possible approaches, allows for natural and flexible distribution of resources aim to provide better performance than best input system. Voting procedures are examples of data fusion – results from identical data sets are merged.

This paper is devised to show the results of MFCC algorithm using density based clustering. This paper is organized as follows: Next we describe prior rela describing MFCC and Db-scan. Section 3 describes the learning of MFCC with Db-scan. Then in section 4, the experiments and evaluation results are explained and discussed finally, section 6 describes the conclusion and future works.

2. Related work

2.1. Multitype Features Coselection for Clustering (MFCC):

In this section we briefly discuss about MFCC. It is made clear that the selection of each type feature and the clustering is an iterative one. After one iteration of clustering, each data object will be assigned to a cluster. In [6], Liu et al. assumed each cluster corresponded to a real class. Using such information, they did supervise feature selection, such as Information Gain (IG) and $\chi 2$ statistic (CHI) [2] during k-means clustering. MFCC tries to fully exploit heterogeneous features of a web page like URL, anchor text, hyperlink, etc., and to find more discriminative features for unsupervised learning. We first use different types of features to do clustering independently. Then, we get different sets of pseudoclass, which are all used to conduct iterative feature selection (IF) for each feature space.

After normal selection, some data fusion methods are used to conduct iterative feature selection (IF) for each feature space, i.e., feature coselection. In each iteration of clustering, the coselections in several spaces are conducted one by one after clustering results in different feature spaces have been achieved before any coselection. Thus, the sequence of coselection will not affect the final performance. The general idea of coselection for k-means clustering is described in fig-1.

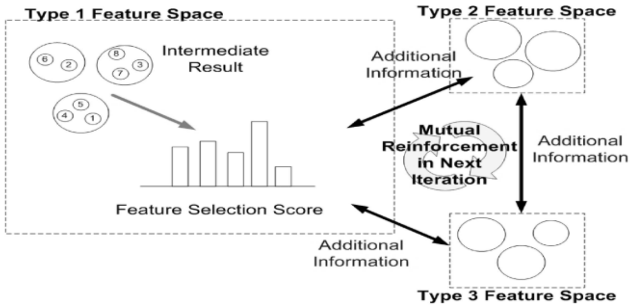


Fig – 1. The basic idea of Multitype feature coselection.

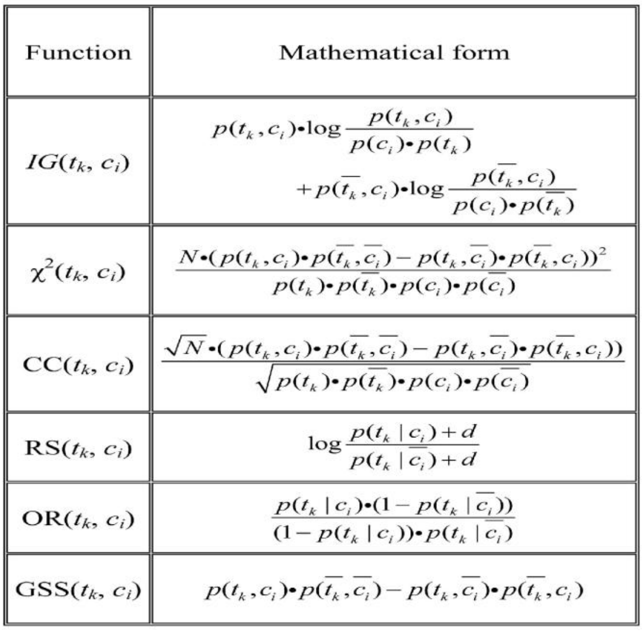
Suppose that we categorize data objects with M heterogeneous features into L clusters. Let fv_n be one dimension of the feature vector, icr_i be the intermediate clustering results in the i^{th} feature space, SF be the fusion function.

The pseudo algorithm is listed as follows:

```
Loop for N iterations of k-means clustering
  {
        Loop for m feature spaces
        {
          Do clustering in feature space m
        }
         Loop for M feature spaces
        {
              For feature space m, do feature selection using
        results in all feature spaces.
         For ( $fv_n$ )  one dimension of the feature vector in space m,
        a feature selection score fss ( $fv_n, icr_i$ ) is obtained by using
        intermediate clustering results $icr_i$ in feature space i.
              Then a combined score fss ( $fv_n$ ) is achieved
        by fusing the scores based on different result sets.

        $fss(fv_n) = SF(fss(fv_n, icr_i))$            (1)
              }
          }
```

In the equation (1), $fss(fv_n, icr_i)$ can be the value calculated by the selection function or rank among all features. The feature selection criteria, the six commonly used feature selection function mentioned in [2]:



Function	Mathematical form
$IG(t_k, c_i)$	$p(t_k,c_i) \bullet \log \frac{p(t_k,c_i)}{p(c_i) \bullet p(t_k)} + p(\overline{t_k},c_i) \bullet \log \frac{p(\overline{t_k},c_i)}{p(c_i) \bullet p(\overline{t_k})}$
$\chi^2(t_k, c_i)$	$\frac{N \bullet (p(t_k,c_i) \bullet p(\overline{t_k},\overline{c_i}) - p(t_k,\overline{c_i}) \bullet p(\overline{t_k},c_i))^2}{p(t_k) \bullet p(\overline{t_k}) \bullet p(c_i) \bullet p(\overline{c_i})}$
$CC(t_k, c_i)$	$\frac{\sqrt{N} \bullet (p(t_k,c_i) \bullet p(\overline{t_k},\overline{c_i}) - p(t_k,\overline{c_i}) \bullet p(\overline{t_k},c_i))}{\sqrt{p(t_k) \bullet p(\overline{t_k}) \bullet p(c_i) \bullet p(\overline{c_i})}}$
$RS(t_k, c_i)$	$\log \frac{p(t_k \mid c_i) + d}{p(t_k \mid \overline{c_i}) + d}$
$OR(t_k, c_i)$	$\frac{p(t_k \mid c_i) \bullet (1 - p(t_k \mid \overline{c_i}))}{(1 - p(t_k \mid c_i)) \bullet p(t_k \mid \overline{c_i})}$
$GSS(t_k, c_i)$	$p(t_k,c_i) \bullet p(\overline{t_k},\overline{c_i}) - p(t_k,\overline{c_i}) \bullet p(\overline{t_k},c_i)$

Table-1 Feature Selection Functions.

Depending on the choices of fss and SF, we obtain five fusion models including voting, average value, max value, average rank, and max rank. The equations are listed as follows:

MaxRank(Rank($f_{\mathbf{vn}}$)) = arg max(Rank(fvn,icr_i))
AverageRank(Rank(fvn))=(ΣRank(fvn,icr_i))/M
Voting(val(fvn)) = Σvote(fvn,icri)
Vote(fvn,icri)={0 val(fvn,icri) <st
                    1 val(fvn,icri) >=st
Average(val(fvn))= Σ val(fvn,icri)/M
Max(val(fvn)) = arg max(val(fvn,icri))

Table-2 Fusion Models

In the above equation, *val(fv_n,icr_i)* is the value calculated by selection function, *RANK(fv_n,icr_i)* is the rank of fv_n in the whole feature list ordered by *val(fv_n,icr_i)*, and st is the threshold of feature selection. After feature coselection, objects will be reassigned, features will be reselected, and the pseudoclass-based selection score will be recombined in the next iteration. Finally, the iterative clustering and feature coselection are well integrated.

In each of the iterations, the whole feature space should be reconsidered. The reason is that our method can help in finding more effective features through a mutual reinforcement process. Properly selected features will help clustering and vice-versa. That is to say, some discriminative features will not be found until late in the clustering phase. This can be proved by empirical results.

2.2 DBSCAN Clustering

DBSCAN regards clusters as dense regions of objects in the data space that are separated by regions of low density. A cluster is defined by this algorithm as a maximal set of density-connected objects. DBSCAN grows regions with sufficiently high density into clusters. Every object not contained in any cluster is considered to be noise.

In DBSCAN for each point of a cluster the neighborhood of a given radius (ε) has to contain at least a minimum number of points *(MinPts)* where ε and *MinPts* are input parameters [11].

The DBSCAN algorithm finds clusters as follows:

Let X = {x_1, x_2, x_3, ..., x_n} be the set of data points. DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a cluster (MinPts).

1) Start with an arbitrary starting point that has not been visited.
2) Extract the neighborhood of this point using ε (All points which are within the ε distance are neighborhood).
3) If there is sufficient neighborhood around this point then clustering process starts and point is marked as visited else this point is labeled as noise (Later this point can become the part of the cluster).
4) If a point is found to be a part of the cluster then its ε neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all ε neighborhood points. This is repeated until all points in the cluster are determined.
5) A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
6) This process continues until all points are marked as visited.

The computational complexity of DBSCAN is *O(NlogN)* if a spatial index is used. Otherwise, it is $O(N^2)$, where *N* is the number of objects. The main advantage of DBSCAN is that it is capable of discovering clusters of arbitrary shape. The main disadvantage of DBSCAN is that it leaves the user with the responsibility of selecting parameter values for e and *MinPts* that will lead to high quality clusters. The quality of the resulting clusters is sensitive to the user-defined parameters.

3. Proposed Work

3.1. Db Scan in MFCC

In this paper we present a method to build a clustering system that merges MFCC with density based clustering. The general idea for modification is based on the coselection and maximal set of *density-connected* objects. Db-scan grows regions with sufficiently high density in to clusters. Advantages of many of Db-scan algorithm include time efficiency and ability to find clusters of arbitrary shapes. MFCC reduces the noise feature effectively by and improved further performance. The modified MFCC got the idea from the arbitrary shapes, where we get intermediate membership to the noise features. So that the selection score for the modified MFCC will be as,

$$fss(\varepsilon, \min pts) = \underset{i=1}{\overset{n}{sf}} \left(\underset{c=1}{\overset{npts}{fss}} (p_{i,c}, \varepsilon, \min pts) \right) \tag{2}$$

whereas,

SF – selection function to fuse the feature space selection (or, the intermediate clustering)

fss – feature selection score to select best center point (or, mean) from the specified feature space.

ε – max. distance between two samples for them to be considered as same in the neighborhood.

minpts – minimum number of points that must exist in the ε neighborhood.

npts – neighborhood points.

p – size of database.

c – cluster.

3.2 Experiments & Results:

The MFCC with Db-Scan algorithm proposed in the paper has been fully implemented and evaluated with extensive experimentation; this section presents the details of implementation, data set and text results.

3.2.1 .Evaluation metrics:

A number of metrics used in feature selection and clustering are evaluated and measures for categorization effectiveness. We use the best recall k precision metrics. Such measures are F-measure and time precision in each fss criteria.

F-measure is calculated by the harmonic mean of vocabulary terms (P) and total terms(R). Each fss criteria define the P & R terms.

We also use accuracy in this paper as a measure. Accuracy is computed as the ration of correctly classified testing documents to the total number of testing documents. Of course, all these performance metrics are computed for each category separately (i.e.) we apply all the testing documents to each fss criteria to compute P, R, f1, and accuracy for each fss criteria.

3.2.2 Experiment Results:

The experimental evaluation was performed on testdata data set. Here we can explain and results on testdata dataset. The testdata contains almost 255 articles, evenly distributed on 10 categories. Further each article can be assigned to one or more clusters. In our experiments run MFCC algorithm having db-scan is tested on above said database.

Density based algorithm – DB-SCAN with MFCC is verified with test data database (Table – 3). It contains feature classes of HTML, text files, word documents, jpeg files, user logs, etc.

Classes	No of documents	Related terms	Total term frequency
ASP	2	22	23
CSS	10	1439	6771
Gif	144	975	976
Html	25	14210	63392
Jpeg	18	3554	3659
Js	19	4935	38415
Pdf	5	249229	398036
Php	10	1670	4644
Png	9	245	245
Ppt	13	193505	208541

Table – 3. Feature Classes of test database.

MFCC algorithm clusters the dataset according to the query term. TF-IDF is calculated and the following result is got for CHI-square ($\Psi^{2)}$, correlation coefficient (CC), GSS coefficient (GSS), and information gain (IG) for each feature class.

DBSCAN is that it is capable of discovering clusters of arbitrary shape. It leaves the user with the responsibility of selecting parameter values for ε and *MinPts* that will lead to high quality clusters. The quality of the resulting clusters is sensitive to the user-defined parameters. (refer fig – 2).

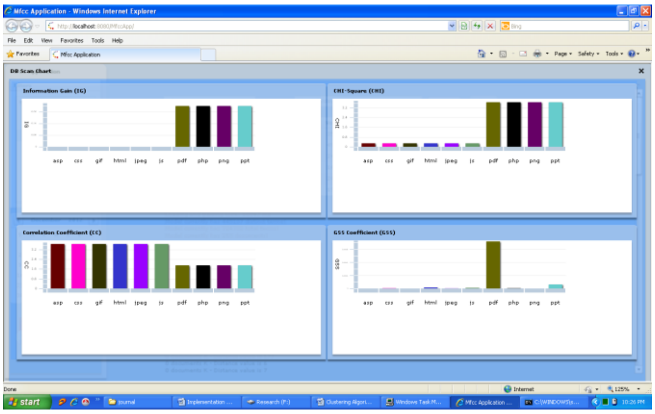


Fig – 2 DB-SCAN.

The testdata database is verified with hard k-means MFCC. The result is shown in Fig-3; the hard k-means clusters

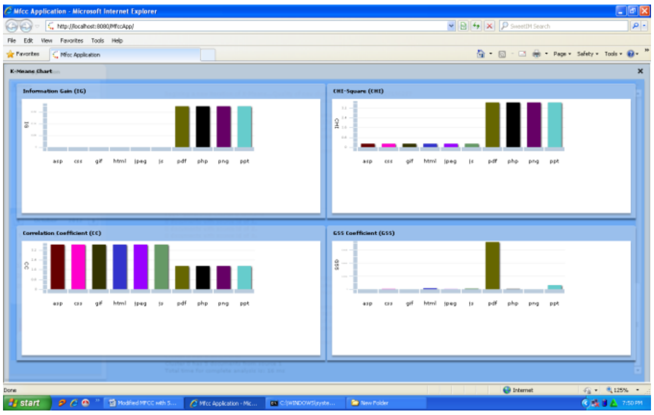


Fig – 3. Hard k-means

The hard k-means and Db-Scan shows the results more or less similar, they differ in time factor, Db-Scan works better in high density data sets. It shows the result in quality of cluster. Since ε and minpts are the two required user defined parameters, need not to specify the number of clusters as opposed to hard k-means. Db-Scan is notion of noise. If the ε neighborhood contains sufficient points then the particular point is marked as noise. Here in MFCC implementation the fss selects the feature space by intermediate clustering and then it fuses the fss score based on data sets. Thus the result is of db-scan and hard k-means is shown in table-4.

Feature selection functions	Cluster model	Number of clusters	Mean values	Time (ms)
IG	Db-scan	10	1,1,2,177,4,177,6,7,177, 177	109
	k-means	2	7.2707	94
Ψ^2	Db-scan	10	1,1,2,177,4,177,6,7,177, 177	15
	k-means	2	7.2707	32
CC	Db-scan	10	1,1,2,177,4,177,6,7,177, 177	15
	k-means	2	11.6937	31
GSS	Db-scan	10	1,1,2,3,4,5,6,7,8,9	0
	k-means	2	3.6349	16

Table – 4. Comparison of Db-Scan and hard k-means

The density based clustering shows better result than hard k-means clustering. Even though the two clusters show the same result, they differ in time factor (fig – 4)

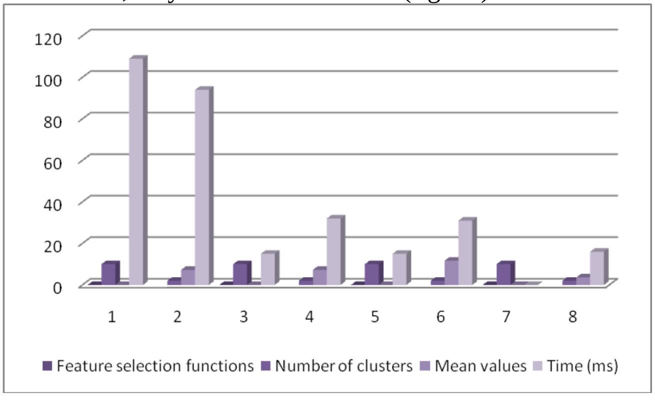


Fig – 4. Comparison of Db-scan & Hard k-me

4. Conclusion

The MFCC algorithm implemented with db-scan shows that it quality clusters and also maintains time criteria. So the higher dimension and high density data set can be clustered and features space can be framed by the fusion methods. The minpts and ε are the parameters determine the cluster shape and quality. The quality of the db-scan is implemented and shown. Db-scan is better in time factor and we need not to specify the number of clusters. Thus we have a chance of discovering further cluster or noise point to be revisited and processed. The MFCC algorithm can be implemented in other clustering algorithms or further extended to other data sets and applications.

References

[1] M. Dash and H. Liu, “Feature Selection for Classification,” Int’l J. Intelligent Data Analysis, vol. 1, no. 3, pp. 131-156, 1997.
[2] Y. Yang and J.O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” Proc. Int’l Conf. Machine Learning (ICML ’97), pp. 412-420, 1997
[3] Shen Huang, Zheng Chen, Yong YU & WeiYing Ma, "Multi type Features Coselection for Web document Clustering", IEEE Transactions on Knowledge and Data Engineering; vol-18,no.4,April2006.
[4] M. Dash and H. Liu, "Feature Selection for Clustering," Proc. 2000 Pacific-Asia Conf. Knowledge Discovery and Data Mining, pp. 110-121,2000.
[5] H.C.L. Martin, A.T.F. Mario, and A.K. Jain, "Feature Saliency in Unsupervised Learning," Technical Report, Michigan State Univ.,2002
[6] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, "An Evaluation on Feature Selection for Text Clustering," Proc. Int'l Conf. Machine Learning(ICML'03), pp. 488-495, 2003.
[7]R. Weiss, B. Velez, M.A. Sheldon, C. Namprempre, P. Szilagyi, A.Duda, and D.K. Gifford, "HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering, "Proc. Seventh ACM Conf. Hypertext, pp. 180-193, 1996.
[8] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training," Proc. Information and Knowledge Management, pp. 86-93, 2000.
[9] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," Proc. Conf. Computational Learning Theory, pp.92-100, 1998.
[10]M. Montague, "Metasearch: Data Fusion for Document Retrieval,"PhD Thesis, Dartmouth College, 2002.
[11]”DB SCAN algorithm” , Wikipedia, the free encyclopedia.

Mrs.K.Parimala.,MCA, Research Scholar in Computer Science in Alagappa University, Karaikudi, INDIA, under the guidance of **Dr.V.Palanisamy**, Professor & Head In-Charge, Department of Computer Science & Engineering, Alagappa University. Currently working as Assistant Professor, in NMS SVN College with a teaching experience of 14 years.

Dr. V.PalaniSamy, MCA, MTech (Adv.IT), Ph.D, Professor & Head In-Charge, Department of Computer Science & Engineering, Alagappa University, Karaikudi, TamilNadu, INDIA, specialized in Algorithms, Wireless Networks & Network Security. He has 20 years of teaching experience and 15 years of Research Experience.