**A REPORT ON**

# **A new feature selection method to improve the document clustering using particle swarm optimization algorithm**

**Submitted To: Dr. M. Venkatesan**

**Submitted By: Harita Reddy (15CO217), Namratha Raj M (15CO132), Naladala Indukala (15CO230), Amshula US (15CO206)**

****

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

**Title:** A new feature selection method to improve the document clustering using particle swarm optimization algorithm

**Aim**

The amount of text documents available on the World Wide Web (WWW) has increased so rapidly in the recent times that dealing such volumes of information has become an extremely complicated task. Text clustering is a commonly used method to deal with such large number of documents by putting them into groups. The paper proposes a feature selection method using the particle swarm optimization (PSO) algorithm (FSPSOTC) to create a new subset of informative text features. Selecting such an informative subset of features improves the clustering performance as well as reduces time complexity.

**Proposed Method and Related Concepts**

The authors’ proposed method FSPSOTC uses particle swarm optimization (PSO) technique to improve text clustering results. The method is divided into 3 stages:

**Stage 1: Text Pre-processing**

In this stage, the authors utilize text-preprocessing methods to convert the text documents into their corresponding numerical representations. The steps involved in this stage are as follows:

1. *Tokenization:* It is the process of splitting a piece of text into tokens or words.
2. *Stop Word Removal:* Stop words are often removed from the text before applying Natural Language Processing (NLP) techniques to the text. Stop words are the commonly used words in English, for instance ‘a’, ‘the’, ‘an’ etc, which have very high frequency of occurrence in text documents. They add little or no value to the content or meaning of the text and their presence can hinder text clustering algorithms from getting a good result.
3. *Stemming:* Stemming transforms appropriate inflectional forms of some words to the same root by removing the prefixes and suffixes of each word.
4. *Weighting Terms:* The term weighting is assigned for each term or feature according to its term frequency in each recorded document.

**Stage 2: Feature Selection Using Particle Swarm Optimization**

This stage uses particle swarm optimization (PSO) to perform feature selection on the vectorized textual features. The PSO Algorithm solves a given problem by generating or having predetermined set of candidate solutions, or here in particular called as particles. These particles move around in the search-space according to some mathematical formulae or constraints over the particle's position and velocity. Every particle move based on its local best known position called LB, and they are also guided to move towards the best known positions in the search-space called the global best GB, which are updated as and when better positions are found by other particles. Through this procedure, this swarm moves towards the best solutions.

The idea behind feature selection to select an optimal subset of features. Suppose the set of textual features for a given document i is defined as Fi = fi,1, …….. fi,t, where t is the number of textual features for the documents. After the selection of informative features, we obtain FSi = si,1, …….. si,m, with a length of m, where si,j is in the range 0 to 1 for j= 1, 2, …., m. If si,j =1, the jth feature is selected as informative feature in the document i. If si,j = 0 , the jth text feature is an uninformative feature in the document i.

In the PSO algorithm, we take the subset of features (that are considered informative) as a candidate solution or a particle. Each particle contains positions, which are nothing but the features. The PSO algorithm begins with random solutions and then tries to find out the most optimal solution. If the jth position is 1, then the jth feature is an informative feature, and if 0, it is not an informative feature. If the jth position is -1, it indicates that the jth feature is not present in the document. The candidate solution is represented as follows:

X 0 1 1 −1 −1 1 0 −1 1 −1

After each generation, all the candidate solutions are evaluated through the fitness function. In this work, the authors use Mean Absolute Difference (MAD) as the fitness function. This measure is used to assign the features a relevance score.

MAD(Xi) = 1/ai \* (|xij - <xi>|)

<xi> = (1/a) \* (xij)

MAD(Xi) is the fitness function of the solution i and xij is the value of the jth feature in the ith document, where the document is in the TF-IDF representation. ai is the number of the selected text features in document i, t is the number of all text features, and <xi> is the mean value of the vector i.

**PSO Algorithm**

**Input:** Generate initial particles randomly

**Output:** Optimal particle and its fitness value

1. Initialize swarm and parameters of PSO including c1, c2 etc.
2. Evaluate all the particles using fitness function.
3. **While** termination criteria, Do
   1. Update the velocity.
   2. Update each position.
   3. Evaluate the fitness function.
   4. Replace the worst particle with the best particle.
   5. Update LB and GB.
4. End **While**
5. Return subset of informative features D’.

The particle swarm optimization memory (PSOM) is initially filled using S randomly generated candidate solutions. With each particle attempting to move the optimal position, the position and velocity of the particles are updated with the following equations

xij = xij + vij

Vij = w\*vij + c1\*rand1\*(LBl - xij ) + c2\*rand2\*(GBl - xij )

Here, w is inertia weight which balances the global exploitation and local exploration abilities of the particle. c1 and c2 are acceleration constants to be initialised before. LBl is the currently best local solution at the lth iteration and GBl is the currently best global solution at the lth iteration. rand1 and rand2 are random numbers in the range 0 to 1. The inertia weight w is calculated using the equation:

w = (wmax - wmin)\* (Imax - Imin) / Imax + wmin

where wmax and wmin are the biggest and smallest inertia weights respectively. These inertia weights fall in the range from 0.5 to 0.9.

Solutions are updated by a discrete value for each dimension. The sigmoid function is used to calculate the probability of the ith position.

sij = 1 if rand < 1/(1 + exp-vij)

0 otherwise

where rand is the random number generated in the range 0 to 1. To update the position, we use

xij = 1 if rand < sij

0 otherwise

An example of the feature selection procedure is shown below.

**Term frequencies of the original features of the documents (D)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | **1** | 0 | 5 | 1 | 0 | **2** | 0 | 0 | 0 |
| 2 | 1 | 0 | 2 | 1 | **5** | 0 | 1 | 0 | 0 | 0 |
| 3 | 3 | 1 | 0 | 2 | 1 | **4** | 1 | 0 | **1** | 1 |
| 4 | 2 | 5 | **4** | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | **3** | 0 | 1 |

The above table shows the terms frequencies of the original features of documents (D). The features selected as uninformative are shown in bold. After the application of feature selection technique, the uninformative features are removed.

**The removed uninformative features in D**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | 0 | 0 | 5 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 3 | 1 | 0 | 2 | 1 | 0 | 1 | 0 | 0 | 1 |
| 4 | 2 | 5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1 |

If the particular feature doesn’t appear in any document, that feature is removed, thus reducing the dimension. Here feature number 9 is removed.

**Subset of informative features D’ (Returned by the PSO Algorithm)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0 | 0 | 0 | 5 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 2 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3 | 3 | 1 | 0 | 2 | 1 | 0 | 1 | 0 | 1 |
| 4 | 2 | 5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 1 |

**Stage 3:**

We get a set of informative features using PSO. We then perform K-means document clustering. Cosine similarity measure is used to compute similarity between document and cluster centroid.

**K-Means Algorithm:**

**Input**: A collection of text documents *D*, and *K* is the number of all clusters.

**Output**: Assign *D* to *K*.

**Termination criteria**

Randomly choosing *K* documents as clusters centroid *C* = (*c*1, *c*2, …, *cK*).

Initialize matrix X as zeros

for all *d* in *D* do

let *j* = *argmaxkϵ*{1 *to K*}, based on *Cos*(*di*, *ck*).

Assign *di* to the cluster *j*, A[*i*][*j*] = 1.

end for

Update the clusters centroid using (where *di* denotes the document number *i* that belongs to cluster centroid number *i*, *akj* is the total number of text documents that belong to cluster *j*, *ri* is the number of text documents in cluster *i)*

**End**

**Results**

The dataset used is: *20Newsgroups* (We will work on a partial dataset with only 4 categories out of the 20 available in the dataset.)

Code is available at: <https://github.com/HaritaReddy/IR_Project>

**Dataset**

Classes of the data: [2 2 2 ... 2 2 1]

Number of Topic Clusters: 4

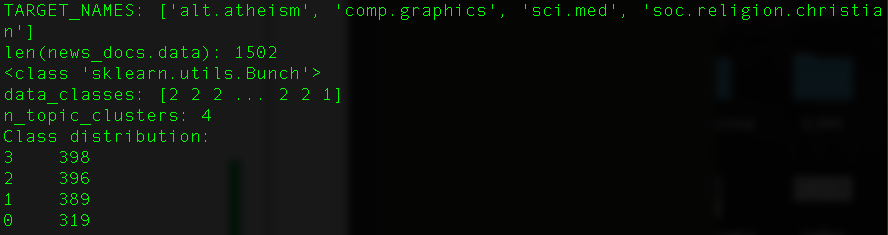
Class distribution:

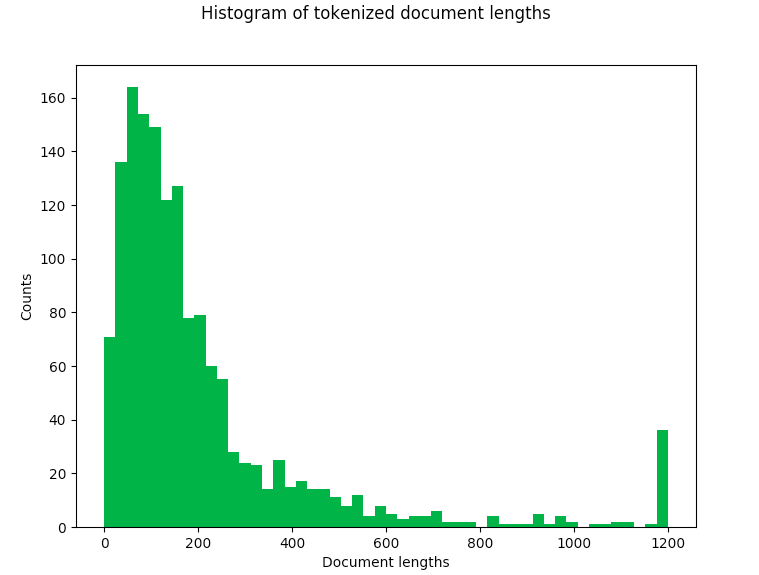
3 398

2 396

1 389

0 319



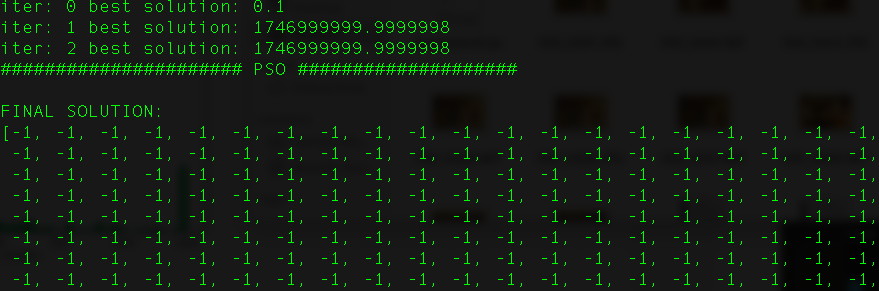


**PSO Iterations**

Iteration 0: best solution: 0.1

Iteration 1: best solution: 1746999999.9999998

Iteration 2: best solution: 1746999999.9999998



**Cluster Centers after K means**

[[-1. -1. -1. ... -1. -1. -1. ]

[-0.9993 -0.9993 -0.9993 ... -0.9985 -0.9993 -0.9985]

[-1. -1. -1. ... -1. -1. -1. ]

[-1. -1. -1. ... -1. -1. -1. ]]