# A PSO-based Clustering Approach Assisted by Initial Clustering Information

Carlos Velázquez, Leticia Cagnina and Marcelo Errecalde

LIDIC. Departamento de Informática. Universidad Nacional de San Luis
Ejército de los Andes 950. (D5700HHW) - San Luis - Argentina.
carvear20@yahoo.com.ar, {lcagnina,merreca}@unsl.edu.ar

**Abstract.** Clustering of short texts is an important research area because of its applicability in information retrieval and text mining. To this end was proposed CLUDIPSO, a discrete Particle Swarm Optimization algorithm to cluster short texts. Initial results showed that CLUDIPSO has performed well in small collections of short texts. However, later works showed some drawbacks when dealing with larger collections. In this paper we present a hybridization of CLUDIPSO to overcome these drawbacks, by providing information in the initial cycles of the algorithm to avoid a random search and thus speed up the convergence process. This is achieved by using a pre-clustering obtained with the *Expectation-Maximization* method which is included in the initial population of the algorithm. The results obtained with the hybrid version show a significant improvement over those obtained with the original version.

**Keywords:** Short-Text Clustering, Bio-Inspired Methods, PSO-based Clustering, Hybrid Methods, Expectation-Maximization, Initialization Approaches.

## 1 Introduction

In recent years, document clustering has become a fundamental process in many tasks as enhancing the results returned by search engines, text mining, unsupervised text organization and information retrieval. In this context, much of the useful information to be processed is taken from Web repositories whose documents are, frequently, short texts with a few tens or hundreds words, such as scientific abstracts, news and short technical and legal documents. For instance, abstracts of scientific papers are often given for free access in most of digital libraries and on line repositories, in opposition to the full texts of the articles. Organizing that huge volume of short texts is an important challenge, as it has been observed in many works on clustering of scientific abstracts [1,2,3].

Several techniques have been developed to solve clustering problems and those based on the *Swarm Intelligence* (SI) paradigm seem to be specially attractive because of their robust performance [4,5,6,7]. One of the main difficulties faced by clustering techniques when applied to collections containing very short documents is the low frequency of terms from text. In this type of domains, an interesting SI algorithm named *Particle Swarm Optimization* (PSO) [8], has been successfully used giving origin to CLUDIPSO, an effective discrete PSO method to cluster short-text corpora [9,10].

On the other hand, it is widely accepted in the clustering research community the importance of providing some initial information to a clustering algorithm. This information can play a key role in helping the algorithm to search the solution space in a more effective way, avoiding local optima and obtaining better quality clustering. These ideas have been applied in different hybrid PSO-based approaches [11,12] but they also have a long tradition in classical clustering approaches like *K-means* whose performance heavily depends on how the initial centroids are distributed [13,14].

In the present article, we extend some preliminary works on CLUDIPSO [9,10], by analyzing how initial information impacts in the performance of this discrete PSO algorithm. This initial information is provided by incorporating in the initial swarm one particle that contains information about the results obtained with other clustering algorithm *Expectation-Maximization* (EM). The results show that the introduction of the named initial information generates a significant improvement in the performance of CLUDIPSO.

The rest of this paper is organized as follows. Section 2 describes the CLUDIPSO algorithm. Section 3 presents the proposal. Section 4 details the experimental study. Section 5 describes the statistical analysis and Section 6 shows the conclusions and future work.

## 2 The CLUDIPSO algorithm

CLUDIPSO is basically a discrete PSO algorithm that works with a population of particles which represent valid clusters. In each iteration of the algorithm two important values are recorded: *gbest* and *pbest*. The first represents the best objective value achieved by a particle of the population, while the second represents the best individual value achieved by a particle along all iterations of the algorithm [8]. The evolution of the particles to find the best solution is performed through the use of two updating equations. Equation (1) represents the change of direction of the *i*-th particle while Equation (2) represents the change of position of the particle.

$$v_{id} = w\,(v_{id} + \Omega 1(pb_{id} - par_{id}) + \Omega 2(pg_{d} - par_{id})) \tag{1}$$

$$par_{id} = pb_{id} \tag{2}$$

Where $par_{id}$ is the value of the particle $i$ at the dimension $d$, $v_{\mathrm{id}}$ is the velocity of the particle $i$ at the dimension $d$, $w$ is the inertia factor (whose goal is to balance global exploration and local exploitation). $\Omega 1$ is the personal learning factor, $\Omega 2$ is the social learning factor, both multiplied by two random numbers generated by a Normal distribution in the range [0,1]. The term $pg_d$ represents the best position reached by a particle in the population and $pb_{id}$ the best position reached by the particle $i$ until that moment.

It is important to note that in CLUDIPSO the updating process is not carried out in all dimensions at each iteration. In order to determine which dimensions of a particle will be updated the following steps are performed:

1. All dimensions of the velocity vector are normalized in the [0,1] range, according to the process proposed by Eberhart and Shi [15] for a discrete PSO version;

2. A random number $r \in [0,1]$ is calculated;

3. All the dimensions (in the velocity vector) higher than $r$ are selected in the position vector and updated using the Equation (2).

To help avoiding convergence to a local optimum, a dynamic mutation operator [16] is used, which is applied to each individual with a *pm*-probability. This value is calculated considering the total number of iterations in the algorithm (cycles) and the current cycle number as the Equation (3) indicates:

$$pm = pm\_max - ((pm\_max - pm\_min)/max\_cycle) * current\_cycle \qquad (3)$$

Where *pm_max* and *pm_min* are the maximum and minimum values that *pm* can take, *max_cycle* is the total number of cycles that the algorithm will iterate, and *current_cycle* is the current cycle in the iterative process. The mutation operation is applied if the particle is the same that its own *pbest* [10]. The mutation operator swaps two random dimensions of the particle.

In CLUDIPSO each valid clustering is represented by a particle, which is a vector of *n* dimensions of integers (*n* represents the number of documents in the collection). Figure 1 shows a valid clustering of *n* documents that were grouped into 3 different clusters.

This version, like that proposed in this paper, uses the Silhouette coefficient (Global Silhouette Coefficient) as a function to be optimized [10]. This coefficient was selected because it has shown a good degree of correlation with the true categorization performed by an expert.

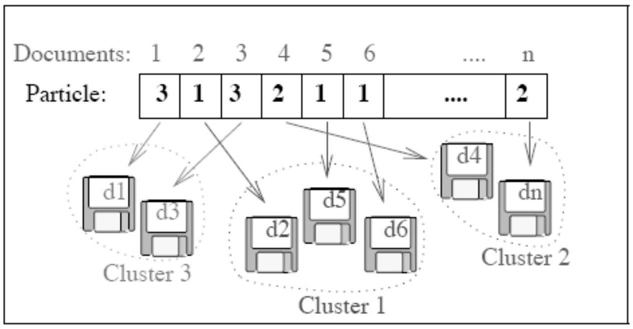

**Figure 1.** CLUDIPSO´s particle representing a valid clustering of *n* documents grouped into 3 clusters.

## 3 The proposal: hybrid CLUDIPSO

The objective of this proposal is to avoid the blind search performed by CLUDIPSO in the first iterations of the algorithm, through the incorporation of information in the initial population. This information is obtained by running a clustering method, which returns with little computational effort, a clustering that represents a particle of the population and thus share information with other individuals. Although there is no guarantee that this clustering is of good quality, it means an important aid to CLUDIPSO. For this task we used the efficient EM method which is implemented in a software platform for machine learning and data mining written in Java and developed at the University of Waikato, Weka [17].

EM corresponds to a family of models known as *Finite Mixture Models*, which can be used to segment data sets. It is a probabilistic clustering method, under which it tries to obtain the probability unknown density function that belongs to the entire data set [18]. Proceeds in two steps iteratively *Expectation*, using the values of the parameters, initial or provided by the step *Maximization* of the previous iteration, obtaining different ways of the probability density function searched. On the other hand the step *Maximization*, which obtains new values of the parameters from the data provided by the previous step [18].

The clustering obtained with the EM method is incorporated into the initial population of particles and the algorithm begins the process of evolution as in the previous version of CLUDIPSO.

## 4 Experimental study

For the experimental study three collections R6, R8B and JRC-Acquis [19] were used. R6 and R8B, are subsets of documents of the collection R8-Test, a subcollection of the dataset Reuters-21578 with news. JRC-Acquis is a subcollection of Acquis, a popular collection of legislative documents of the European Union. Table 1 shows the number of documents (|*DOC*|), the number of terms (|*T*|) and the number of groups for each collection (|*G*|).

**Table 1.** Characteristics of the collections used.

| Collection | \|DOC\| | \|T\| | \|G\| |
|---|---|---|---|
| R6 | 536 | 53494 | 6 |
| R8B | 816 | 71842 | 8 |
| JRC-Acquis | 563 | 1424074 | 6 |

For the EM algorithm were used Weka default settings which include only five iterations to obtain a valid clustering.

CLUDIPSO was executed with the following parameters: 50 particles (one of which initially contains the grouping obtained with EM), 10000 iterations, factors of personal and social learning *Ω1* and *Ω2* set at 1.0, *pm_min* = 0.4, *pm_max* = 0.9 and the inertia factor *w* = 0.9. For each experiment were performed 30 independent executions in order to obtain statistically comparable results. These parameters were selected based on [20].

The quality of the results was evaluated and compared through the use of the classic external measure of quality F-Measure. These results are shown in Table 2. Values highlighted in bold indicate the maximum and minimum best values obtained for each collection considered.

Table 2 shows that hybrid CLUDIPSO obtained the best maximum values for all collections and in some cases, with a notable difference with respect to those of CLUDIPSO. Similar results can be observed with the minimum values for which clearly hybrid CLUDIPSO outperforms CLUDIPSO. These results demonstrate the good performance that hybrid CLUDIPSO obtained for larger collections of short texts.

**Table 2.** F-Measure values obtained with each algorithm.

| Algorithm | Minimum | Maximum |
|---|---|---|
| CLUDIPSO (Collection: R6) | 0,26 | 0,38 |
| Hybrid CLUDIPSO (Collection: R6) | **0,48** | **0,51** |
| CLUDIPSO (Collection: R8B) | 0,18 | 0,25 |
| Hybrid CLUDIPSO (Collection: R8B) | **0,37** | **0,42** |
| CLUDIPSO (Collection: JRC) | 0,26 | 0,33 |
| Hybrid CLUDIPSO (Collection: JRC) | **0,50** | **0,55** |

## 5 Statistical analysis

In recent years, the use of statistical tests to improve the process of evaluating a new method has become a crucial and necessary task in the field of computational intelligence. Usually, they are used within the framework of an experimental analysis to decide when an algorithm is considered better than another. This task, which could be not trivial, it has become necessary to confirm whether a proposed new method offers a significant improvement on existing methods or not, for a given problem [21].

To analyze statistically the distribution of data of the original version [20] and hybrid CLUDIPSO, boxplots [22] were performed with the values obtained in 30 independent runs. Boxplots display graphically differences between samples (results of the experiments) using a box (the size indicates the data dispersion), divided by a segment between 25 and 75 percentiles, the median. The vertical lines outside the box indicate smallest and largest observations (whiskers) and outliers are represented by the symbol '+'. Figure 2 displays boxplots obtained for each algorithm with each collection.

Figure 2 shows the boxplots for the version of CLUDIPSO original and hybridized for the collection R6. Both boxplots are at different heights indicating the difference of performance obtained by each algorithm. There is also a difference between the sizes of the boxplots, indicating that the original version has greater dispersion than the hybridized version.

With respect to the collection R8B, the boxplots in Figure 2 indicate the difference between the performance of CLUDIPSO and the hybridized version, visibly favoring the latter. With respect to the distribution of data, the boxplot of CLUDIPSO presents a slightly lower dispersion than hybrid CLUDIPSO.

For JRC-Acquis collection shown in Figure 2, we note that the height difference between both boxes is the most significant of the three comparisons. Regarding the size of the boxplots, these are similar, indicating similar data dispersion. With respect to data distribution in the case of the hybrid algorithm is strongly biased to the right, indicating that the data tend to concentrate toward the bottom of the distribution.

As a final conclusion of this study of statistical distribution, we can state that significant improvement is visible not only in the maximum and minimum values of F-Measure but remains adequate dispersion of data distribution between both versions of CLUDIPSO.

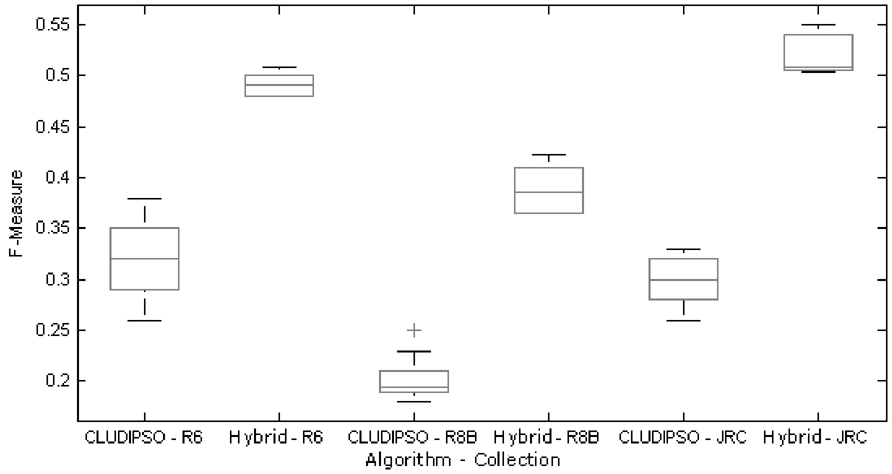

**Figure 2.** Data distribution and collection for each algorithm.

## 6 Conclusions and Future Work

This paper proposed a hybridization of the algorithm CLUDIPSO to improve the performance of its predecessor, when it is used to cluster larger collections of short text.

Results of previous works on CLUDIPSO indicate its effectiveness in small collections of short texts, but not in those of larger (more than 50 documents).

The hybridization involved the incorporation of information generated by a powerful and recognized clustering algorithm (EM) in the initial population of CLUDIPSO. We evaluated the performance of the proposal with three collections R6, R8B and JRC-Acquis, achieving better results than its predecessor without hybridizing.

By studying statistically the distribution of data, it was concluded that in most cases, the dispersion of data is similar for both versions.

As future work, we will make a comparison between hybrid CLUDIPSO and the algorithm CLUDIPSO* which presents good evidence when working with larger collections. Also, we will make a comparison between these results obtained and the experiment with hybrid CLUDIPSO, but without the use of the mutation, to see if better results are achieved with less computational complexity.

## References

1. Alexandrov, M., Gelbukh, A., Rosso, P., An Approach to Clustering Abstracts. In: Montoyo, A., Muñoz, R., Métais, E. (eds.) Natural Language Processing and Information

Systems, Lecture Notes in Computer Science, vol. 3513, pp. 1-10. Springer Berlin / Heidelberg. (2005).

2. Makagonov, P., Alexandrov, M., Gelbukh, A., Clustering Abstracts Instead of Full Texts. In: Proc. of Int. Conf. on Text, Speech and Dialogue, TSD 2004, Lecture Notes in Artificial Intelligence, vol. 3206, pp. 129-135. Springer-Verlag. (2004).

3. Errecalde, M., Ingaramo, D., Rosso, P., Proximity Estimation and Hardness of Short-Text Corpora. In: Proc. of 5th Int. Workshop on Text-Based Information Retrieval, TIR 2008, pp. 15-19. IEEE CS. (2008).

4. Bin, W., Zhongzhi, S., A Clustering Algorithm Based on Swarm Intelligence. In: Proc. of the Int. Conf. on Info-tech and Info-net, ICII 2001. vol. 3, pp. 58-66. (2001).

5. Labroche, N., Monmarché, N., Venturini, G., AntClust: Ant Clustering and Web Usage Mining. In: Genetic and Evolutionary Computation Conf. pp. 25-36. Chicago. (2003).

6. Xiao, X., Dow, E., Eberhart, R., Miled, Z.B., Oppelt, R., Gene Clustering using Self-Organizing Maps and Particle Swarm Optimization. In: Proc. of the 17th Int. Symposium on Parallel and Distributed Processing. (2003).

7. Krink, T., Paterlini, S., Differential Evolution and Particle Swarm Optimization in Partitional Clustering. Computational Statistics and Data Analysis 50, pp. 1220-1247. (2006).

8. Shi, Y., Eberhart, R., A Modified Particle Swarm Optimizer. In: Proc. of the IEEE Int. Conf. on Evolutionary Computation. pp. 69-73 (1998).

9. Ingaramo, D., Errecalde, M., Cagnina, L., Rosso, P., Computational Intelligence and Bioengineering, Chap. Particle Swarm Optimization for Clustering Short-Text Corpora, pp. 3-19. F. Masulli and A. Micheli and A. Sperduti Eds. IOS Press (2009).

10. Cagnina, L., Errecalde, M., Ingaramo, D., Rosso, P., A Discrete Particle Swarm Optimizer for Clustering Short-Text Corpora. In: Int. Conf. on Bioinspired Optimization Methods and their Applications, BIOMA 2008. pp. 93-103 (2008).

11. Cui, X., Potok, T.E., Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm. The Journal of Computer Science pp. 27-33 (2005).

12. Omran, M., Salman, A., Engelbrecht, A., Dynamic Clustering Using Particle Swarm Optimization with Application in Image Segmentation. Pattern Analysis & Applications 8, pp. 332-344. (2006). http://dx.doi.org/10.1007/s10044-005-0015-5, 10.1007/s10044-005-0015-5

13. Bradley, P., Fayyad, U., Refining Initial Points for K-means Clustering. In: Proceedings of the Fifteenth International Conference on Machine Learning. pp. 91-99. ICML '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998).

14. Maitra, R., Peterson, A., Ghosh, A., A Systematic Evaluation of Different Methods for Initializing the K-means Clustering Algorithm. IEEE Transactions on Knowledge and Data Engineering (2010).

15. Hu, X., Eberhart, R., Shi, Y., Swarm Intelligence for Permutation Optimization: a Case Study on n-queens Problem. In Proc. of the IEEE Swarm Intelligence Symposium, pp. 243-246. (2003).

16. Cagnina, L., S. Esquivel, Gallard, R., Particle Swarm Optimization for Sequencing Problems: a Case Study. Congress on Evolutionary Computation, pp. 536-541. (2004).

17. García Jiménez, M., álvarez Sierra, A., Análisis de Datos en Weka. Pruebas de Selectividad. http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf

18. Garré, M., Cuadrado, J., Sicilia, M., Rodríguez, D., Rejas, R., Comparación de Diferentes Algoritmos de Clustering en la Estimación de Coste en el Desarrollo de Software. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 3, N° 1. (2007).

19. Errecalde, M., Recursos de Procesamiento de Lenguaje Natural. https://sites.google.com/site/merrecalde/resources

20. Cagnina, L., Ingaramo, D., Errecalde, M., Performance Analysis of Particle Swarm Optimization Applied to Unsupervised Categorization of Short Texts. Procesamiento del Lenguaje Natural, N° 47, pp. 207-214. (2011).

21. Derrac, J., García, S., Molina, D., Herrera, F., A Practical Tutorial on the use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. Elsevier, Swarm and Evolutionary Computation. (2011).

22. Tukey, J. W., Exploratory Data Analysis. Addison-Wesley Publishing Company, Reading, MA. (1977).