Particle swarm optimization for support vector clustering Separating hyper-plane of unlabeled data

Souad Chaabouni, Salma Jamoussi, Yassine Benayed

Particle swarm optimization for support vector clustering

Separating hyper-plane of unlabeled data

Souad CHAABOUNI

Multimedia, InfoRmation systems and
Advanced Computing Laboratory,
MIRACL.
SFAX University
Sfax, Tunisia
chaabouni_souad@yahoo.fr

Salma JAMMOUSSI

Multimedia, InfoRmation systems and
Advanced Computing Laboratory,
MIRACL.
SFAX University
Sfax, Tunisia
jamoussi@gmail.com

Yassine BENAYED

Multimedia, InfoRmation systems and
Advanced Computing Laboratory,
MIRACL.
SFAX University
Sfax, Tunisia
yassine.benayed@gmail.com

Abstract— **The objective of this work is to design a new method to solve the problem of integrating the Vapnik theory, as regards support vector machines, in the field of clustering data. For this we turned to bio-inspired meta-heuristics. Bio-inspired approaches aim to develop models resolving a class of problems by drawing on patterns of behavior developed in ethology. For instance, the Particle Swarm Optimization (PSO) is one of the latest and widely used methods in this regard. Inspired by this paradigm we propose a new method for clustering. The proposed method PSvmC ensures the best separation of the unlabeled data sets into two groups. It aims specifically to explore the basic principles of SVM and to combine it with the meta-heuristic of particle swarm optimization to resolve the clustering problem. Indeed, it makes a contribution in the field of analysis of multivariate data. Obtained results present groups as homogeneous as possible. Indeed, the intra-class value is more efficient when comparing it to those obtained by Hierarchical clustering, Simple K-means and EM algorithms for different database of benchmark.**

Keywords—Support Vector Machines, clustering, Particle Swarm Optimization

I. INTRODUCTION

The Support Vector Machines (SVM) appeared for over fifty years and was proposed by Vapnik [1]. They constitute one of the few learning methods that are completely derived from theoretical considerations. Therefore it is mathematically well-founded. Indeed, these large margins separators are based on two key ideas, allowing the processing of nonlinear discrimination problems, and the reformulation of the classification problem as a quadratic optimization problem. Moreover, these methods allow the pooling of the maximum margin concept and the notion of kernel function. SVM are supervised classification techniques based on the principle of structural risk minimization rather than empirical risk.

Thanks to its theoretical basis and to benefit from its used cores, we are very interested to exploit the advantages of SVM for the problem of data clustering. Indeed, the enormous quantities of information gathered from various sophisticated technologies such as computers and satellites, have been born an urgent and insistent need to intelligent tools for the analysis, management and organization of this bulky information. In another words, the clustering term is associated with the notion of abstraction. It is used to summarize information in groups most generals. Indeed, a clustering aims to bring together a set of observations in homogeneous classes by promoting heterogeneity between these different classes. In this context, the general principle of clustering consists of minimizing the distance between two individuals of the same class and maximizing the distance between two individuals of distinct classes.

Besides in recent years, the living world can offer methods to problem-solving that are inspired by its mechanisms. Indeed, there are varieties of research in the literature showing the ability of bio inspired algorithms to produce good results for complex and large problems, such as clustering. The particle swarm optimization is a new bio-inspired optimization method that presents simplicity of algorithms, a convergence speed and excellent results.

Then originally, the method PSvmC (Particle of SVMs for Clustering) was created specifically to address the problem of using SVM in a different field of classification. This problem, which is the main application of this paper, is a clustering problem.

In this paper, first of all we introduce the various work using SVMs in the clustering field. Then, we present a detailed description of our algorithm, its formal presentation and the various experiments performed.

II. RELATED WORK

The main idea of SVMs is that these methods were developed to solve a classification problem (in two classes) by computing an optimal class separator. The separating hyper-plane is calculated by projecting the training set defining the two classes in a higher dimensional space. The classification of each data point will be done according to its position to the separator.

In other terms, the optimal hyper-plane is the hyper-plane whose minimum distance to the training examples is maximal: Maximize margin subject to classifying all points correctly.

The Fig.1 shows the different linear hyper-plane allowing the separation of data and the optimal hyper-plane ensuring the maximization of the margin.
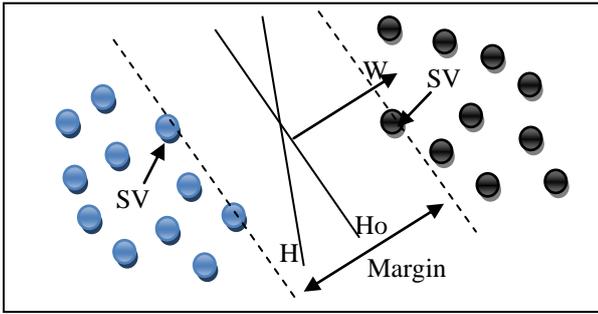

Fig.1. Separation of two sets of points by linear separators

Solving the problem of calculating the optimal hyper-plane is done in a completely mathematical and theoretical manner (more details in [2]). Thanks to the advantages of SVM methods and their solid theoretical foundation, several researchers have attempted to use it in the clustering field but in a semi-supervised way. Indeed, there are few adaptations for pure unsupervised case.

The most famous clustering algorithm based on SVMs is the one invented by Vapnik and called SVC (support vector clustering). We will detail the SVC principles in the following of this section. Also we will make an overview of the hybridization attempts of SVMs with other clustering methods (hierarchical method or k-means method).

- SVC basing on the approach of SVMs, [3] has created a novel clustering method which identifies clusters by the separation of each one with a set of contours. This method proceeds by mapping the data points in a high dimensional feature space with the support of a Gaussian kernel. Afterwards, it searches the smallest sphere that encompasses the data image being in the feature space. Finally, it ensures the mapping back of the found sphere to the data space. That creates a set of contours which enclose the data points. Cluster designs points enclosed in the same contour.

- Support Cluster Machines: This method is proposed by [4] to effectively deal with large-scale classification problems. The main idea of the SCM is to cluster the training set and then apply the SVM classification method on clusters previously obtained.

- CB-SVM: The classifying large data sets using SVMs with hierarchical clusters (using the algorithm BIRCH) presents a semi supervised classification method proposed by [5]. It allows the recursive selection of the clusters centroids as the representatives along the hierarchical clustering tree.

- The cluster-based data selection method [6], presents a cluster-based learning methodology to reduce training time and the memory size for SVM. Using the Kmeans clustering technique, a set of cluster is created and then filtered. A set of critical data is so prepared to be classified with the SVM method.

III. DESCRIPTION OF THE PSvmC METHOD

PSvmC 'Particle of SVMs for Clustering' is our proposed algorithm. It allows the combination of the PSO meta-heuristic [7], with the benefits of SVM: the supervised classification method for clustering purposes.

PSvmC provides grouping of data in two different classes. Indeed PSvmC is an algorithm able to create a separator of data without labellum. To do this, the swarm, turned by the PSvmC algorithm, presents a collection of particles. Their number is well determined. Each of these particles consists of defining an optimal hyper-plane separating the data into two classes. These particles move in the data space, keeping in mind the best separator already found, with a velocity properly adjusted. The moving of all particles of swarm led to discovery a global solution of the problem of finding the best separating of data. Hereafter we will detail the main elements of the PSvmC method.

A. Score function

In our work, the measurement of the performance quality of the solution is made through a fitness function to maximize. We seek to choose an objective function which estimates the group quality that we want to refine and improve. In the literature, the internal validity indices such as the inertial indices are the most objective functions used by the famous clustering algorithms, this is explained by the computing quality and performances offered by these indices.

Therefore, after a study of different criteria that ensure a good clustering [2], we choose to use inertial index: Intra-class inertia, because it is the simplest and most effective measure used for calculating the raw similarity between objects of same class. Furthermore, the computation time of the intra-class inertia is reasonable and the noisy data does not produce a calculation problem. This index is in a digital format that we want to minimize:

$$I_{intra} = 1/n \times \Sigma\Sigma \ d^2(x,g_i) \qquad (1)$$

Where n presents the number of elements and g_i is the gravity center of the class i.

To receive the benefits of this measure we should made a clear expression of distance between objects. We choose the Manhattan distance because firstly we have sets of multivariate data and secondly this measure is not sensitive to noisy data. Indeed, the Manhattan distance calculates the sum of absolute values of differences between the coordinates of two points:

$$d \ (X, \ Y) = \Sigma \ | \ x_i - y_i | \qquad (2)$$

B. Particles

The particle notion originating of nature presents a simulation of bees or birds that use suitably chosen paths in their problem of foraging. A particle is then moving, and it has a velocity. Also, each particle has a small memory, which allows remembering its best performance and position. Also, it

has a group of informants, generally called its neighborhood. In our method, we consider that the particles are characterized by four features:

- A current separating hyper-plane presents the current position of the particle.

- Current intra-class inertia of data which describes the performance value, also called the objective function at the current position.

- The best hyper-plane and its best intra-class inertia, previously found by the particle itself. This is the simulation of the memory characterizing a particle.

- The best hyper-plane and its best intra-class inertia, discovered by the entire swarm. This is to emphasize the neighborhood topology chosen.

To better explain the concept of particle, we will detail in the following: the position, the objective function, the particle velocity and the rules for updating velocity and mutations submitted to renovate the particle swarm.

C. Particle position

In this work, the particle position presents a hyper-plane formed by an array of support vectors which are generated by the SVM method. The support vectors "SV" are the data samples that maximize the distance or the margin separating the positive samples from the negative ones, or simply separate the two classes of the database. The support vector synthesizes someway the important aspects of the sample set. In fact, only the examples corresponding to support vectors are actually useful in learning. If we knew them a priori, we could make learning without considering of other examples. Briefly, in our program we used model files which characterize hyper-planes, theoretically written as follows,

$$h(x) = \omega(x) + b, \tag{3}$$

Where ω is the weight vector $\omega = (\omega_1 \ldots \omega_n)$ and b is a constant. A model file contains a set of support vectors and some parameters such as the constant that specifies the maximum-margin between the data-points of the two classes noted b or rho.

A particle at a certain position is a set of SV at a given iteration. This means that a movement of a particle causes a change of the SV set of the hyper-plane.

D. Particle velocity and its updating rules

Let V_i^t, be the velocity of the particle i at iteration t. The new velocity of i is calculated as a linear combination of three elements by using confidence coefficients:

- The adventurous tendency, i.e. continue under the current speed of the current position "$current_{position}$".

- The conservative tendency bringing more or less towards the best position already found "$best_{particle}$".

- The tendency of overreaction, pointing towards the best informant position of the swarm "$best_{neighbor}$".

The application of the combination of these three elements to the current position gives the new position of the particle. Formally the equation of velocity is written as follows with c_1, c_2 and c_3 are the confidence coefficients where their sum equals to 1.

$$V_i^{t+1} = c_1 \times V_i^t + c_2 \times (best_{particle} - current_{position}) + c_3 \times (best_{neighbor} - current_{position}) \tag{4}$$

As we have defined the position of particle by the set of support vectors, the velocity at a given iteration will be defined as the combination of a number of support vectors taken from the current position "currentSV", a number of support vectors taken from the best position already found "pBestSV" and a number of support vectors taken from the best position achieved by the swarm "gBestSV".

The simulation of the subtraction ($best_{particle} - current_{position}$) (respectively $best_{neighbor} - current_{position}$) returns the support vectors belonging to the pBestSV (respectively gBestSV) and outside of the currentSV. As we are currently managing sets, the subtraction will be written as the set of pBestSV (respectively gBestSV) private of the currentSV set:

pBestSV \ currentS V (respectively gBestSV \ currentSV).

The velocity is then written as follows:

$$V_i^{t+1} = c_1 \times currentSV + c_2 \times (pBestSV \setminus currentSV) + c_3 \times (gBest \setminus currentSV) \tag{5}$$

In our approach, the correlation coefficients represent the percentage of retained SVs from the result of the previously subtraction operation. The value of these coefficients is not arbitrarily chosen, it follows a definite rule, which we will detail in the following:

- Rules for updating the velocity:

The random selection of the correlation coefficients needs a new parameter describing the maximum velocity of particle which must not be exceeded because generally, a high velocity causes a poor exploration of the search space. Our idea to simplify and minimize the number of parameters to be determined, leads us to create a new rule for determining the correlation coefficients. The latter are determined automatically depending on the intra-class inertia of each particle position. As the constraint:

$$c_1 + c_2 + c_3 = 1 \tag{6}$$

So c_1, c_2, c_3 must belong to] 0, 1[, and knowing that inertia is a positive and strictly greater to 1 so its inverse satisfies the constraint and presents a value between 0 and 1. To avoid being trapped in very small values and to normalize the inverse of the inertia, we divided it by a common divisor which presents the sum of the inverses of all intra-class inertia "Ia": That is to say the sum of the inertia inverse of the $current_{position}$ position, the inertia inverse of the $best_{particle}$ position and the inertia inverse of the $best_{neighbor}$.

$$Divisor = \frac{1}{1/Ia_{currentSV} + 1/Ia_{pBestSV} + 1/Ia_{gBestSV}} \quad (7)$$

Correlation coefficients are then written as follows:

$$c_1 = \frac{1/Ia_{currentSV}}{Divisor}$$

$$c_2 = \frac{1/Ia_{pBestSV}}{Divisor}$$

$c_3 = 1 - (c_1 + c_2)$; (for satisfying the constraint (6)).

Calculating these coefficients ensures that the coefficient value increases when the particle position has much more interesting performance. So, when a position has minimal intra-class inertia, the coefficient value will be maximized and then the particle can take as much information as possible from this position.

- Rules for updating the position

After calculating the new velocity, each particle considers this information to his new movement. Indeed, we multiply each correlation coefficient by the total number n_{tot} of support vector of the new position. So the new position consists of $n_{tot} \times c_1$ support vectors from the current position, $n_{tot} \times c_2$ support vectors from the best particle position and $n_{tot} \times c_3$ support vectors from the best neighborhood position. As a certain position is composed by a set of support vectors presenting the crucial and decisive samples of the two-class data, the choice of a K number of support vectors in a given position arise the following question:

- How many SV must be chosen from each class?

To answer this question, we used the same rule for determining the correlation coefficients seen above but by using the intra-class inertia on each class. The number of SV selected from each class i, is:

$$N_i = n_{tot} \times c_k \times \frac{1/I_i}{1/I_1 + 1/I_2} \quad (8)$$

With $k = 1 \dots 3$ and I_i presents the intra-class inertia of each class i.

The selection of N_i support vectors of class i, is done by a completely random manner.

E. Mutation operator

To prevent a premature convergence of a search algorithm, the evolutionary programming paradigm introduces an original concept called mutation. The introduction of a mutation operator provides our method with a better exploration capacity of the search space. Thus, the algorithm can converge to the global optimum. This operator consists in changing the current position randomly with a probability equal to 2%. The mutation operator of the algorithm PSvmC embodies the type of deletion mutation. Indeed, it removes 10% of support vectors of a given position.

F. The swarm

The collective behavior of particles defined above leads to the evolution of the swarm which tries to reach interesting sites. Indeed, the random initialization of the particles positions with uniform distribution provides an excellent exploration of the search space and then more chance to converge to the global optimum.

To further explain this choice, the use of the random criterion allows obtaining good and bad position of particles which ensures the separation of data. In other terms, we allow a great value of intra-class inertia. This variety of quality can have a significant gain in terms of solutions variability. This promotes a convergence towards the correct solution beyond the local optimality.

G. Number of particles needed in the swarm

The quantity of particles assigned to solve a problem depends on two parameters:

- The size of the search space.

- The relationship between the computing capacity of the machine and the maximum search time.

We have proposed to determine the particle number automatically depending on the attribute number of the database. Indeed, if N is the attribute number, the size of the swarm is given by:

$$Nbr_{particle} = (K \times N) \quad (9)$$

With K represents the determination coefficient of the particles number.

Theoretically, a study is needed to extricate the value of the coefficient K. For this, we performed experiments on a set of two-class databases (see the next sections). For each value of the coefficient K, the program of PSvmC algorithm was run 10 times by returning the value of intra-class inertia. To determine the value of the coefficient K we analyzed the development of the average of the 10 intra-class inertia relative to different values of K: 1, 2, 3… We did get the results shown in Fig.2, Fig.3. (See all experiments in [2]).

N iter	AVR intra
1	5504.386
2	5576.912
3	4768.9
4	4746.828
5	4900.828
6	4676.14

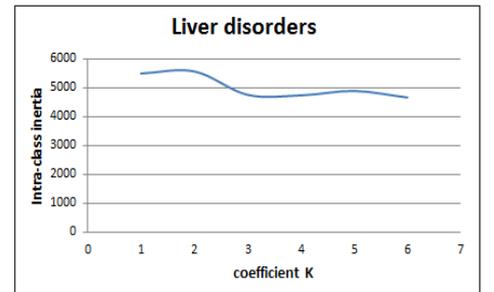Table 1: Various tests of inertia calculation.

Fig.2. Effect of the particle number on the intra-class inertia made on the Liver Disorders database.

For the Liver Disorders database, since the value K = 3, the intra-class inertia values are limited in the interval [4676; 4900]. This presents a good value of intra-class inertia. Also in

treating Heart database, we note that the intra-class inertia is better since the value 3 of the coefficient K.

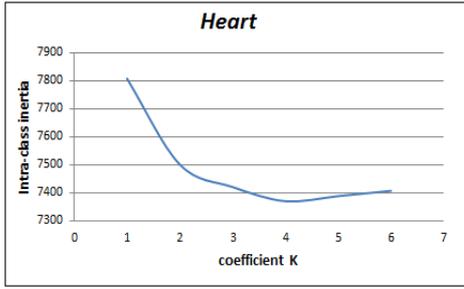| Nbr iter	AVR intra
1	7808.8
2	7502.53
3	7422.362
4	7372.1
5	7389.88
6	7409

Table 2: Various tests of inertia calculation.

Fig.3. Variation of the intra-class inertia of the Heart database according to the particle number.

As the analysis of these results presents that the values of the intra-class inertia shows good stability starting from the value K = 3 for the different databases of benchmark treated, the number of particles as well adopted is $Nbr_{particle} = (3 \times N)$.

IV. FORMAL PRESENTATION OF THE PSVMC ALGORITHM

The global PSvmC process is based on three steps: initialization, update, and displacement movement. These steps are crucial to the process of the PSvmC algorithm. This process is summarized by the Fig.4.
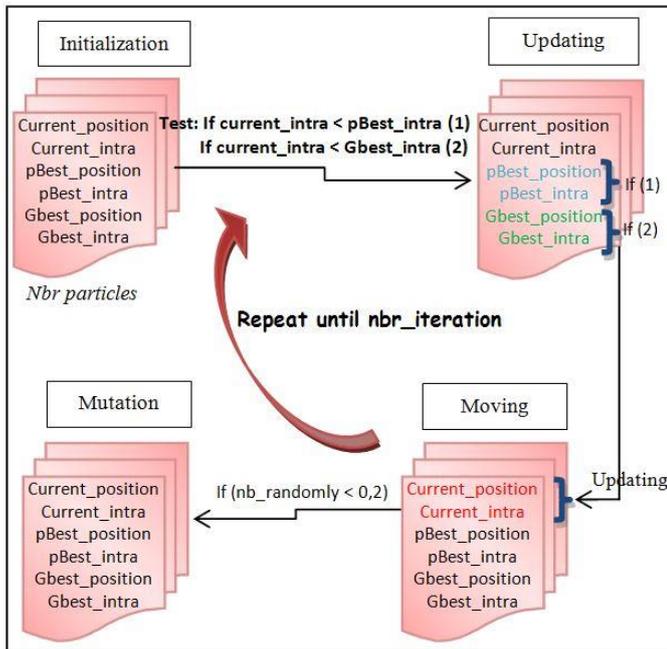

Fig.4. Global processing of the PSvmC algorithm.

During the updating step, each particle of the swarm memorizes the current position and its intra-class inertia if this latter is smaller than its better inertia even found (pBestposition). Besides, the global position (Gbest-position) is updated by the current position if the constraint number (2) presented in Fig.2 is satisfied. The moving and the mutation steps are applied as we have mentioned in the above section.

---

[1] UCI is available in the following web address: http://archive.ics.uci.edu/ml/

[2] LIBSVM is available in the following web address: http://www.csie.ntu.edu.tw/cjlin/libsvm/

The different steps of our method are described in the Algorithm 1:

Algorithm 1 PSvmC Algorithm

Require :$\{x_i\}$: set of unlabeled data.
Ensure :$\{S\}$: separator of the two data groups.
  For each particle i *do*
      Initialize its $Current_{position}(i)$ position with labels randomly taken and its personal best position $best_{particle}$
      Calculate the intra-class inertia to affect it to: $Ia_{currentSV}$ and $Ia_{pBestSV}$.
  Initialize $best_{neighbor}$, the best global position.
  Calculate $Ia_{gBestSV}$, the intra-class inertia of the global position.
  repeat
  for each particle i **do**
    if ($Ia_{currentSV} < Ia_{pBestSV}$) **then**
      Update $best_{particle}$, the best personal position.
      Update the $Ia_{pBestSV}$ inertia.
    if ($Ia_{currentS V} < Ia_{gBestSV}$) **then**
      Update $best_{neighbor}$, the best global position.
      Update the $Ia_{gBestS V}$ inertia.
  for each particle i **do**
    Moving the "$Current_{position}$" current position of the i particle.
    if (Number randomly selected < 0:2) **then**
      Mutation applied on the "$Current_{position}$".
  until The maximum number of iterations is reached.
  return "$best_{neighbor}$" position.

VI. EXPERIMENTS AND RESULTS

A. Used data sets

The data sets used for testing our approach are readily made available by the University of California at Irvine [UCI][1] and the SVM library LIBSVM[2] that contain the most commonly used databases. In our simulations we used a set of different two-class databases, designed for simple supervised classification.

- Pima Indians Diabetes (**1**): used since 1990, this database presented in eight attributes, helps to diagnose and to determine whether a patient shows any sign of diabetes (500 objects) or a good health (268 objects).

- Liver disorders (**2**): presents a set of individuals who complains of a liver disorder. This database contains 345 instances, each one is presented by seven attributes and there are 145 positive and 200 negative instances.

- Heart (**3**): it is made available by LIBSVM. It contains 270 objects described by 13 attributes and containing two classes (120 instances presents a heart disease and 150 instances shows a healthy cardiovascular system).

- Breast cancer (**4**): this database which contains 683 instances (444 positives and 239 negatives) and characterized by nine attributes, describes if a woman has breast cancer, benign or malignant.

- Australian (**5**): it concerns the requests of Australian credit card. It presents a good mix of continuous, nominal attributes with small number of nominal values, and with a greater number of values. It is described by 14 attributes (383 negative, 307 positive).

B. Evaluation

In this section, we describe the experiments that were performed to illustrate the performance of the proposed algorithm PSvmC. To evaluate our work, we made experiments on different data sets, by varying their sizes and comparing the proposed approach to three other algorithms based on different clustering methods: The hierarchical clustering method, the simple k means clustering algorithm and the probabilistic method "EM".

These different grouping methods are available at the Environment for Knowledge Analysis: WEKA [8]. To evaluate the different generated clusters, we adopted the intra-class inertia to measure the similarity between the same class data as evaluation criteria. We launched our proposed algorithm ten times for each experiment and we calculate at each time the inertia intra-class average. Thus we calculate the intra-class inertia of the three others clustering methods by using a good setup. In fact, if we put a wrong parameter of class number, the intra-class value of these methods becomes incomparable with our method.

Database	Clustering methods				
	Hierarchical clustering	*Simple K-means*	*EM*	*PSvmC*	
				Avg	*Min*
1	31489.705	29470.27	47224.17	27000.05	24063.95
2	5965.49	4302.5	3887.445	4518.15	3799.20
3	8857.64	7712.36	8165.44	7604.12	7431.07
4	476.18	178.64	221.22	178.51	176.07
5	13346147.8	27445338	27495486.3	12784926.68	11381873.87

Table3: Comparison of intra-class inertia obtained on the 'Pima Indians Diabetes'(**1**), the 'Liver disorders'(**2**), the 'Heart' (**3**), the 'Breast-Cancer'(**4**) and the 'Australian' (**5**) databases by different clustering methods.

According to the results of the four algorithms presented in the table above (Table3) and tested with the different databases of benchmark, we can deduce various observations. Indeed, we noted that the clusters found by PSvmC have more interesting results than those found by other algorithms. In effect, for the "Pima Indians Diabetes" database, our approach provides a far smaller inertia (24063.95) than those obtained by the other three algorithms. Also, for the database of "Liver Disorders", the table 3 shows that by using PSvmC algorithm the intra-class inertia (3799.20) is also smaller than other methods. For the "Heart" database, we see that the difference between the average of the intra-class inertia and the minimum of the intra-class inertia is low; this shows the stability of our approach throw different executions. Finally, we note that the intra-class inertia found by our approach is more efficient when comparing it to those obtained by Hierarchical clustering, Simple K-means and EM algorithms for the "Breast cancer" database (176.07) and for the "Australian" database.

To explain these obtained results: since we actually define a good clustering as the clustering having the small value of intra class inertia, we turned our algorithm in order to find the smallest value of intra-class inertia. A review of the results of various experiments ensures that we have reached our initial goal.

To conclude, the PSvmC algorithm provides a grouping ensuring a smallest intra-class inertia value than the other existing algorithms. Or, the aim of clustering data is to find small intra-class inertia within the same group to increase the similarities between its elements; we can say that our proposed algorithm achieves better results than the K-means, Hierarchical and EM algorithms.

VII. CONCLUSION

In this paper, we introduced our clustering algorithm called PSvmC allowing the creation of a data separator based on the SVM principles. In effect, the particle swarms are a pertinent source of inspiration for integrating SVM in the field of clustering. Moreover, we could propose a heuristic based on PSO, which aims to solve the problem of knowledge extraction.

We described the various experiments carried out to release the best of our approach and this by the right choice of the parameter values of our algorithm. In addition, several tests were conducted to evaluate the performance of our method using intra-class inertia. A comparison on PSvmC was conducted face the Hierarchical clustering, the Simlpe Kmeans and the EM algorithms.

Thanks to the obtained encouraging results we have the idea of proposing other extensions of the PSvmC algorithm that are a top-down hierarchical clustering and a multi-objective clustering. Also, other prospect is related to the improvement of the presented approach such as the determination of the different adjustment parameters of the SVM methods. Indeed, the best setting may improves over the clustering results. Another perspective is the parallelization of the proposed approach to take maximum advantage of the PSO concept that ensures the parallelization of the optimization search.

References

[1] V. Vapnik, The nature of statistical learning theory. Springer Verlag, 1995. Vii.

[2] S. Chaabouni. Particle swarm optimization for support vector clustering, Master thesis's, Higher Institute of Computer Science and Multimedia Sfax, Tunisia, 2012.

[3] A. Ben-Hur, D. Horn; Hava T. Siegelmann and V. Vapnik. Support Vector Clustering. Journal of machine learning research, vol. 2, pp. 125–137, 2001.

[4] B. Li, M. Chi, J. Fan and X. Xue. Support cluster machine. Machine Learning, volume 227, pp. 505–512, 2007.

[5] H. Yu, J. Yang and J. Han. Classifying Large Data Sets Using SVMs with Hierarchical Clusters. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 306–315, New York, USA, 2003.

[6] S. Sun, C. Tseng, Y. Chen, S. Chuang and H. Fu. Cluster-based Support Vector Machines in Text-Independent Speaker Identification. Proceedings of the Int'l Joint Conf.on Neural Network, 2004.

[7] K. James, P. Riccardo and T. Blackwell. Particle swarm optimization. Journal of Swarm Intelligence, vol. 1, no. 1, pp. 33–57, 2007. 38

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten. The WEKA data mining software: an update. Special Interest Group on Knowledge Discovery and Data mining Explorations, vol. 11, pp. 10–18, 2009.