Information Sciences 339 (2016) 353–368

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A Rough–Fuzzy approach for Support Vector Clustering

Ramiro Saltos*, Richard Weber

Department of Industrial Engineering, FCFM, Universidad de Chile, República 701, Santiago de Chile, Chile

ARTICLE INFO

Article history:
Received 13 November 2014
Revised 17 December 2015
Accepted 25 December 2015
Available online 8 January 2016

Keywords:
Fuzzy sets
Rough sets
Support Vector Clustering
Data mining

ABSTRACT

Support Vector Clustering (SVC) is an important density-based clustering algorithm which can be applied in many real world applications given its ability to handle arbitrary cluster silhouettes and detect the number of classes without any prior knowledge. However, if outliers are present in the data, the algorithm leaves them unclassified, assigning a zero membership degree which leads to all these objects being treated in the same way, thus losing important information about the data set. In order to overcome these limitations, we present a novel extension of this clustering algorithm, called Rough–Fuzzy Support Vector Clustering (RFSVC), that obtains rough–fuzzy clusters using the support vectors as cluster representatives. The cluster structure is characterized by two main components: a lower approximation, and a fuzzy boundary. The membership degrees of the elements in the fuzzy boundary are calculated based on their closeness to the support vectors that represent a specific cluster, while the lower approximation is built by the data points which lie inside the hyper-sphere obtained in the training phase of the SVC algorithm. Our computational experiments verify the strength of the proposed approach compared to alternative soft clustering techniques, showing its potential for detecting outliers and computing membership degrees for clusters with any silhouette.

1. Introduction

Clustering is one of the most important data mining tasks. Its objective is to find natural groups in a given data set in which the observations would be homogeneous within each group and heterogeneous between groups. Many clustering algorithms have been proposed in the literature [10,14,27,34,36,37], which can be grouped into two categories: hard clustering, and soft clustering. Their main difference is that in hard clustering an object has to belong exactly to one cluster while this constraint is relaxed in soft approaches. In certain applications, the hard clustering approach may not be adequate given the nature of the problem. Consequently, soft clustering could grant more flexibility in deriving adequate solutions.

Since their introduction, fuzzy sets [40] and rough sets [26] have shown their particular advantages when ambiguity and uncertainty have to be dealt with [27]. Fuzzy C-Means [5] and Rough C-Means [20] are very common representatives of soft clustering algorithms, and their derivatives have been applied in many areas. However, their use is still limited by some characteristics, such as clusters with spherical shapes, the fact that the sum of the membership values of an object has to be equal to 1 (Fuzzy C-Means), the need to know the number of clusters beforehand, and that the data points identified as outliers are not classified accordingly.

* Corresponding author. Tel.: +56 9 54362841.
E-mail addresses: ramiro.saltos@ing.uchile.cl, rjsaltos1989@gmail.com (R. Saltos), rweber@dii.uchile.cl (R. Weber).

http://dx.doi.org/10.1016/j.ins.2015.12.035

On the other hand, Support Vector Clustering [4], an important density-based clustering algorithm, uses the support vector machines philosophy to find the clusters' cores, gaining the capability of detecting classes of any shape without having to know in advance the number of clusters. However, this algorithm treats all data points that do not belong to one of the clusters found in the same way, indicating them just as outliers. Especially, in the presence of scattered data this turns out to be a major drawback since important information could be lost by not differentiating among outliers.

To overcome these deficiencies, we propose a novel clustering algorithm called Rough–Fuzzy Support Vector Clustering, a generalization of Support Vector Clustering, that as will be shown, has the following advantages: similar to traditional SVC, any cluster shape can be detected and it is not necessary to know the number of clusters in advance, since the basic idea relies on the concept of density of data points. Additionally, those data points that are not clearly assigned to one of the clusters (here considered as outliers) get membership values to all clusters according to their distances in the higher-dimensional feature space. These membership values provide important information about the outliers, which in many applications could be the most critical cases.

The remainder of this paper is arranged as follows: Section 2 presents the traditional Support Vector Clustering algorithm and provides an overview of the state-of-the-art of its soft computing variations. Section 3 introduces the proposed method called Rough–Fuzzy Support Vector Clustering and explains its basic ideas in detail. Experimental results using RFSVC, and alternative methods, are presented in Section 4. Finally, Section 5 contains a summary of this paper, provides its main conclusions, and indicates future developments.

2. Literature overview on Support Vector Clustering

In this section, we present a general introduction to Support Vector Clustering. Then, we provide the respective algorithm's mathematical description in order to have the basis for developing our rough–fuzzy clustering method. Finally, we comment on recent studies related to our approach to emphasize its importance.

2.1. General introduction to Support Vector Clustering

Let $X = \{\mathbf{x}_i \in \mathcal{R}^d / i = 1, 2, \ldots, N\}$ be the set of N data points and $\mathcal{R}^d$ be the data space. The traditional Support Vector Clustering algorithm groups the elements in set X into clusters, interpreting the solution of the Support Vector Domain Description [33] as cluster cores, and assigns each individual point to its nearest core to generate the final clusters [9]. This is achieved using a two-phase algorithm consisting of a *training phase* and a *labeling phase*.

During the training phase, following the ideas proposed by Tax and Duin [33], data points are projected from the original data space to some higher-dimensional space looking for the hyper-sphere with a minimal radius that encloses most of the data points, as shown in Fig. 1(a). When the enclosing sphere is found, three kinds of data points can be identified: support vectors (SV), bounded support vectors (BSV), and inside data points (ID). Support vectors are data points whose images lie on the surface of the enclosing sphere, while bounded support vectors lie outside the hyper-sphere, and inside data points belong to its interior.

After that, the images of data points are projected back from the higher-dimensional space to the original data space where support vectors now define a set of contours that enclose data points. This completes the training phase (Fig. 1(b). Finally, the labeling phase identifies the different clusters found during training and allows building a {0, 1}-membership matrix which indicates to which cluster each data point belongs.

2.2. Mathematical description of Support Vector Clustering

In this section, following Ben-Hur's work, we present the mathematical description of the training phase and the labeling phase of Support Vector Clustering algorithm. For more details and proofs, see [4].

2.2.1. Training phase

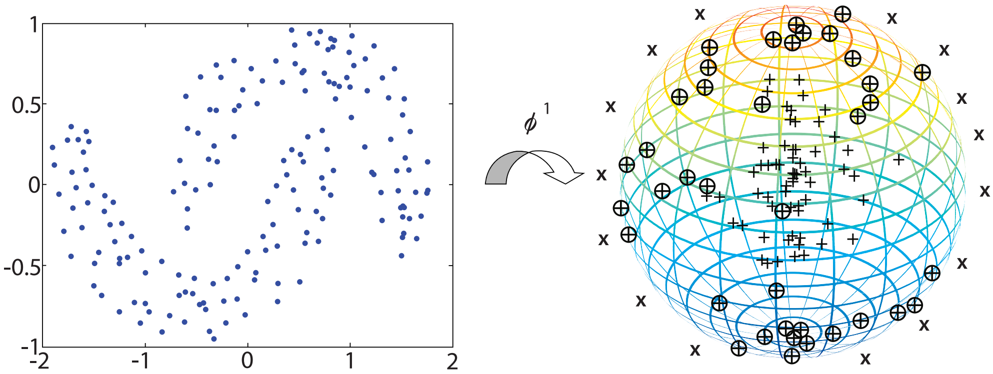
In the training phase, a quadratic optimization problem is solved in order to find the hyper-sphere with minimal radius in a higher-dimensional space that encloses the images of the available data points from the original space. The model is formulated as follows:

$$Min\ R^2 + C\sum_{i=1}^{N} \xi_i \tag{1}$$

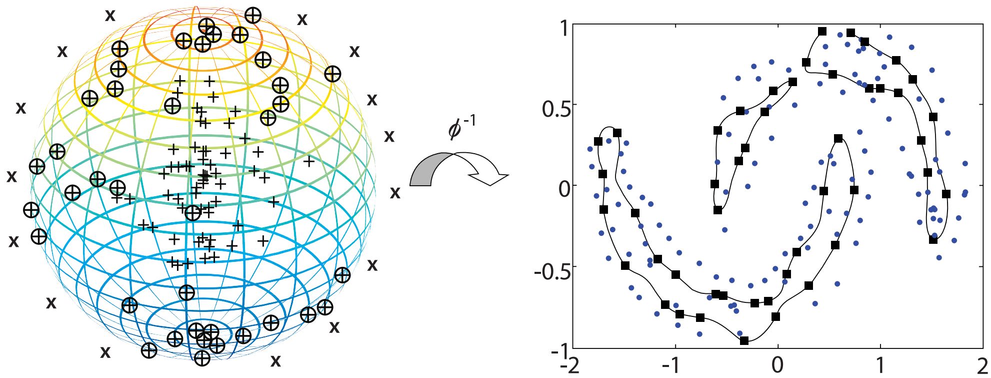
$$s.t.\ \ \| \phi(\mathbf{x}_i) - \boldsymbol{a} \|^2 \le R^2 + \xi_i \quad \forall i = 1, \ldots, N \tag{2}$$

$$\xi_i \ge 0 \quad \forall i = 1, \ldots, N \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm, $\boldsymbol{a}$ is the center of the hyper-sphere, ϕ is the non-linear function that projects data from the original space to the higher-dimensional space, ξ_i are slack variables that relax the constraints to allow some data points to lie outside the sphere, R is the sphere's radius, and $C \in [0, 1]$ is a constant penalty parameter.



(a) Projection from Original Data Space to Higher-Dimensional Space



(b) Inverse Projection from Enclosing Sphere to Cluster Contours

Fig. 1. General idea of SVC (+: inside data points, ⊕: support vectors, **x**: bounded support vectors).

The solution of the primal problem can be obtained by solving its Wolfe dual form that is a function of variables β_i:

$$Max \sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

$$s.t. \sum_{i=1}^{N} \beta_i = 1 \tag{5}$$

$$0 \le \beta_i \le C \quad \forall i = 1, \ldots, N \tag{6}$$

where β_i are Lagrange multipliers and $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ is the kernel function. It can be shown [4] that only those points with $0 < \beta_i < C$ lie on the surface of the hyper-sphere, and are called support vectors (SV) which define the contours that enclose the points in the data space. Data points with $\beta_i = C$ lie outside the sphere and are called bounded support vectors (BSV); the corresponding data points in the original space do not belong to any cluster found. Points with $\beta_i = 0$ lie inside the hyper-sphere and are called inside data points (ID). These are the objects that belong to one of the clusters built.

One widely used kernel function is the Gaussian kernel, which has the following form:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2} \tag{7}$$

with width parameter q. As the value of q increases, the numbers of SVs and clusters increase as has been shown in [4].

For each data point $\mathbf{x}$, the distance of its image in the higher-dimensional space from the center of the hyper-sphere is given by:

$$\begin{aligned} R^2(\mathbf{x}) &= \| \phi(\mathbf{x}) - \boldsymbol{a} \|^2 \\ &= K(\mathbf{x}, \mathbf{x}) - 2\sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{x}) + \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{8}$$

Then, the radius of the hyper-sphere that encloses data points in feature space can be calculated as follows:

$$R_S = \frac{1}{|SV|} \sum_{\mathbf{x}_i \in SV} R(\mathbf{x}_i) \tag{9}$$

Finally, the contours that enclose the points in data space are defined by the set:

$$\{\mathbf{x}/R(\mathbf{x}) = R_S\} \tag{10}$$

The contours are interpreted as forming cluster boundaries as is shown in Fig. 1(b) [4]. In view of this, support vectors lie on cluster boundaries; bounded support vectors are outside; and all other points lie inside the clusters.

In the training phase, the Support Vector Data Description model is used to obtain the support vectors and many efficient algorithms exist to solve the quadratic optimization problem involved [21]. For example, the Generalized Sequential Minimal Optimization (GSMO) algorithm, proposed by Keerthi and Gilbert [13], is widely used to solve optimization problems related to Support Vector Machines.

The training phase is governed by two parameters: q, the scale parameter of the Gaussian kernel, and C, the soft margin constant. Both affect the shape of the enclosing contours in data space. A complete computational study of their effects had been given in [4].

2.2.2. Labeling phase

Although the training phase of the SVC algorithm provides the set of support vectors, bounded support vectors, and inside data points, it does not differentiate between two points that belong to two different clusters. To do so, Ben-Hur et al. [4] propose the following strategy: given a pair of data points, $\mathbf{x}_i$ and $\mathbf{x}_j$, which belong to different clusters, any path that connects them must exit from the sphere in feature space, i.e., $\exists\lambda \in [0, 1]$, such that $R(y_{i,j}) > R_S$, where $y_{i,j} = \lambda\mathbf{x}_i + (1-\lambda)\mathbf{x}_j$.

This leads to the definition of the adjacency matrix A with elements $a_{i,j}$ between pairs of points $\mathbf{x}_i$ and $\mathbf{x}_j$ whose images lie in or on the sphere in the higher-dimensional space:

$$a_{i,j} = \begin{cases} 1 & \text{if } \forall\lambda \in [0,1], R(y_{i,j}) \le R_S \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Clusters are now defined as the connected components of the graph induced by A. Note that bounded support vectors are unclassified by this procedure since the higher-dimensional space images lie outside the enclosing sphere. We can decide either to leave them unclassified, or to assign them to any clusters using some criteria.

The labeling rule introduced by Ben-Hur is called Support Vector Graph [4]. Due to its high computational time [28], other labeling methods have been proposed in the literature such as Proximity Graph Modeling [38], Stable Equilibrium Point [15], Cone Clustering Labeling [16,17], Fast and Stable Labeling [18], among others.

2.3. Extensions of Support Vector Clustering

In this section, we review and comment on the extensions of the Support Vector Clustering algorithm relevant to our work. First, we show soft computing variations and then more recent crisp approaches.

Fuzzy versions of Support Vector Clustering have been presented by Chiang and Hao [6], Zheng et al. [41] and García et al. [11]. In [6], multiple spheres were used to represent each cluster in the higher-dimensional space and a cell-growing method was employed to calculate membership degrees for the data, and to obtain a prototype of each cluster.

In García et al. [11], subtractive clustering [7] was used to obtain the class center of each cluster; then support vector machines (SVM) for density estimation were used; support vectors were found; and the membership degrees for the elements in the clusters were calculated based on the idea of Fuzzy C-Means, i.e. in an iterative fashion. These approaches provide a membership degree matrix, i.e. a fuzzy partition of the data. Finally, Zheng et al. [41] proposed assigning a membership degree to each data point in order to weigh its relevance for classifier construction, i.e. fuzzy information was used at the input level. Despite the fact that Zheng et al. use such membership values, the partition of the data set obtained by their work, i.e. the respective output, is not fuzzy.

One important rough variation of the Support Vector Clustering algorithm was introduced in 2005 by Asharaf et al. [2]. Its key idea is to look for two hyper-spheres that enclose all data points in the higher-dimensional space instead of just one. The data points enclosed by the smaller hyper-sphere are supposed to be elements of the lower approximation of a cluster, while the data points enclosed by the larger hyper-sphere and not enclosed by the smaller one are assumed to be elements of the boundary of the same cluster. Labeling is partially performed by a modification of Ben-Hur's labeling algorithm. With this approach, a soft partition of the data set is obtained, but no membership matrix for the elements in the boundary of the clusters is provided, as will be the case in our approach. Table 1 summarizes the state of the art and main characteristics of these extensions.

More recent studies [8,9,18,28–30] deal primarily with the main two disadvantages of the traditional Support Vector Clustering algorithm, i.e. the computational times used in the training phase and in the labeling phase. Table 2 shows which issue is addressed and whether the respective approach finds a crisp, fuzzy, or rough partition of the data set.

Clearly, major efforts are being dedicated to the labeling phase since this is still a critical issue of SVC. And Table 2 reveals that recent studies do not consider soft computing approaches; another indication of the importance of the work presented in this paper. Finally, Li and Ping [19] presented a complete survey of the literature related to Support Vector Clustering recently, highlighting the benefits, drawbacks, improvements, variations, and research directions of Ben-Hur's algorithm.

Table 1
Soft computing variations of Support Vector Clustering.

Author	Year	Partition	Membership matrix	Prototype description
Chiang and Hao [6]	2003	Fuzzy	Yes	Centers of the spheres in feature space
Asharaf et al. [2]	2005	Rough	No	No prototype
Zheng et al. [41]	2006	Crisp	No	No prototype
García et al. [11]	2006	Fuzzy	Yes	Centers found by substractive clustering

Table 2
Recent improvements for SVC.

Author	Year	Partition	Training phase	Labeling phase
Ping et al. [28]	2010	Crisp	Yes	Yes
Chonghui and Fang [8]	2011	Crisp	Yes	Yes
Ping et al. [30]	2012	Crisp	No	Yes
Ping et al. [29]	2013	Crisp	Yes	Yes
D'Orangeville et al. [9]	2013	Crisp	No	Yes
Li [18]	2013	Crisp	No	Yes

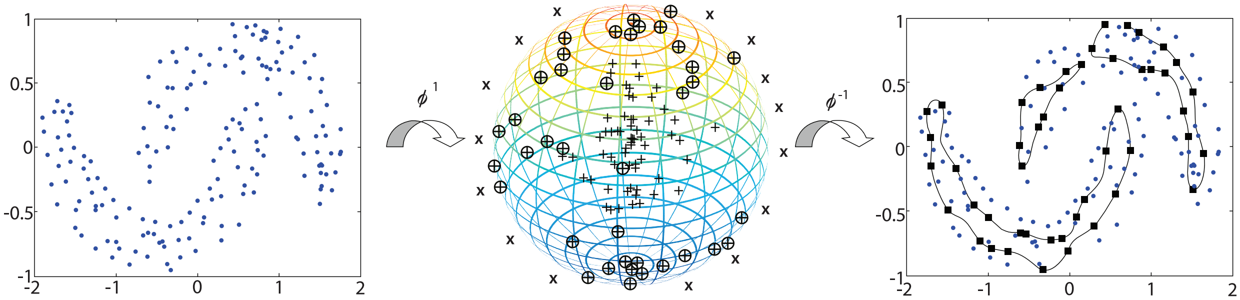


Fig. 2. Data points outside the sphere in the higher-dimensional space.

3. A Rough–Fuzzy approach for Support Vector Clustering

In this section, we present a general overview of our Rough–Fuzzy Support Vector Clustering method first. Subsequently, we explain its details in Section 3.2, putting special emphasis on the fuzzy boundary which is our main contribution. Finally, Section 3.3 shows how the proposed method works via a comprehensible two-dimensional data set which allows visual inspection of each of its steps.

3.1. Overview of the proposed method

During the training phase of traditional SVC, we solve the model defined by Eqs. (1)–(3) as described in Section 2.2.1. If we select $C = 1$ in this model, no data point lies outside the hyper-sphere, so the boundary will be empty and a crisp partition of the data set will be found (Fig. 1). On the other hand, if $C < 1$, some of the data points (BSV) will lie outside the hyper-sphere and will belong to the surroundings of the clusters (Fig. 2). Data points that lie inside the sphere will be elements of the clusters built by the labeling rule used.

The crisp cluster structure generated by traditional SVC will be replaced by rough–fuzzy clusters, characterized by two main components: a lower approximation formed by the elements inside the hyper-sphere and a fuzzy boundary containing the bounded support vectors, i.e., those data points that lie outside the hyper-sphere.

As mentioned in Section 2.2, existing methods for Support Vector Clustering do not classify bounded support vectors (BSV). To do so, we propose a fuzzy boundary for these BSVs, which allows extracting additional knowledge about the data. This is motivated by the following two ideas:

- Some bounded support vectors are closer to one cluster in data space than to others. As a consequence, they should be treated differently.
- Fuzzy set theory provides a mathematical framework for obtaining numerical membership values of the data points that belong to cluster boundaries.

Combining these ideas, we developed a methodology which allows data analysts to have more insights regarding bounded support vectors; e.g., a BSV could be identified as an outlier if it is far away from all the clusters [31].

This provides more information on the nature of bounded support vectors. For example, a BSV with degrees of membership close to 0 to all clusters could be considered to be an outlier while other BSVs could be treated in a different way.

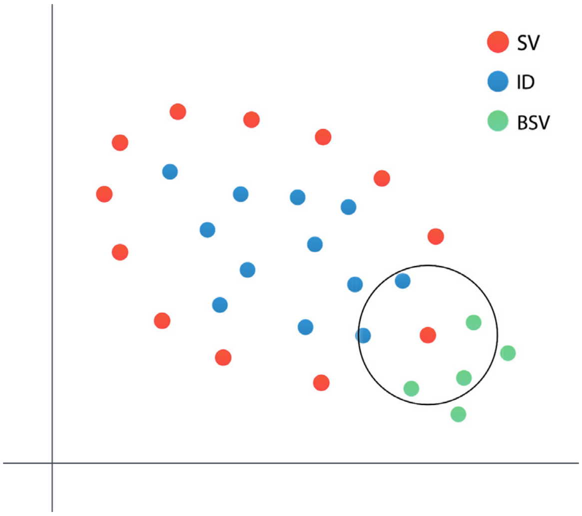


Fig. 3. Support vector and its zone of influence.

This is possible because our approach does not require that the membership degrees must sum up to 1 as in other fuzzy clustering algorithms. This is one of the most important and powerful characteristics of our proposal. Another important advantage of our algorithm is that it is not necessary to assume any prior knowledge about the silhouette and the number of clusters in the data set; i.e. the flexibility of the SVC algorithm is not affected. Additionally, the use of our procedure, after having run the SVC algorithm, is not expensive computationally.

3.2. Development of the proposed method

The outputs of the traditional SVC algorithm are: the set of support vectors (SV), bounded support vectors (BSV), and inside data points (ID), together with the number of clusters and the {0, 1}-membership matrix of the data set in each cluster found. Note that bounded support vectors are unclassified so their membership in each cluster is zero.

As mentioned in Section 2.1, all inside data points lie in the hyper-sphere and will be classified in one of the clusters built by the SVC algorithm. Since the membership of these elements in a specific cluster is not dubious, we postulate that they define the lower approximation of their respective cluster.

The SVC algorithm, however, does not classify the BSV, thus leaving the membership of these elements in any cluster dubious. In order to overcome this issue, we propose that the BSV data form new fuzzy sets.

The construction of the fuzzy boundary is based on the idea that each clusters support vectors could be seen as the centers of ball-shaped clusters with the respective SV as its prototype. The main advantage of this idea is that some BSV will be under the influence of such SV (Fig. 3), which is the best representative of the data close to it.

Using this idea and the distance of a BSV from all SVs in the cluster, we can calculate the respective membership degrees. Given that the cluster structure formed by the SVC algorithm is based on SVs, we must decide which strategy to use to calculate the membership degree of a BSV: considering the distance to the nearest SV, or the average distance to all SVs in the same cluster.

We propose calculating the membership degree $\mu_{i,j}$ of bounded support vector i to support vector j using the Gaussian kernel function as the membership function (Eq. 12) because this kernel maps to (0, 1]. It gets close to 1 when the data points are close to the respective SV and tends to 0 when they are far away.

$$\mu_{i,j} = \mu(x_i, SV_j) = K(x_i, SV_j) = e^{-q\|x_i - SV_j\|^2} \tag{12}$$

We use this kernel function as a membership function because it maintains the same order of the data space in the higher-dimensional space, i.e., support vectors that are close in the data space will be close on the surface of the hyper-sphere in the higher-dimensional space as has been shown in the following theorem:

Theorem 1. *Let x_i, x_j and x_k be elements of a data space $\mathcal{R}^d$. Let $\|\cdot\|$ be the Euclidean norm and $d(\cdot, \cdot)$ be the Euclidean distance. Let ϕ be a non-linear transformation from $\mathcal{R}^d$ to some higher-dimensional space and $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) = e^{-q\|x_i - x_j\|^2}$ be the Gaussian kernel function with width parameter $q > 0$. Then:*

$$d(x_i, x_j) \le d(x_i, x_k) \Leftrightarrow d(\phi(x_i), \phi(x_j)) \le d(\phi(x_i), \phi(x_k))$$

Proof. See Appendix A. □

This important fact allows us using the kernel function as a membership function which is a more natural way to compute the membership degrees than using arbitrary membership functions.

According to the criterion to calculate the membership matrix, the steps to obtain the final membership degree of each BSV are as follows:

- If we consider the nearest SV criterion, the membership degree of BSV_i to cluster j is the value of $K(BSV_i, SV_j)$ where SV_j is the nearest support vector in the cluster j to BSV_i.
- If we consider the average distance to all SVs in the same cluster, we must use Eq. 13.

$$\mu_{i,j} = \frac{1}{|SV_j|} \sum_{\mathbf{x}_k \in SV_j} K(BSV_i, \mathbf{x}_k) \tag{13}$$

where SV_j is the set of support vectors of the cluster j.

Once the idea explained above is applied, we obtain the membership matrix U where the element $\mu_{i,j}$ is the membership degree of bounded support vector i to cluster j. One advantage of this matrix over the matrix provided by Fuzzy C-Means is that the sum of the membership degrees in each row of the former is not necessarily 1. This fact enables our approach to detect outliers. For example, as we will explain in Section 4.2, an outlier will have a membership degree close to zero in all clusters while this value will be significantly different from zero for non-outlier data points.

To complete our rough–fuzzy clustering method, we structure all ideas mentioned in Section 2.2 together with the ideas explained in this section in an algorithm that allows computing the final rough–fuzzy clusters and the membership matrix. This algorithm has been called Rough–Fuzzy Support Vector Clustering (RFSVC) and is shown in Algorithm 1.

Algorithm 1: Rough–Fuzzy Support Vector Clustering.

```
Input: Data set X, parameters q > 0 and υ ∈ (1/N, 1)
Output: Rough–fuzzy clusters with [0, 1]-membership matrix and the number of clusters c
1  Calculate penalty constant C = 1/(Nυ)
2  Run the training phase of the SVC algorithm and obtain the set of support vectors (SV), bounded support vectors
   (BSV) and inside data points (ID).
3  Run the labeling phase of the SVC algorithm and obtain the crisp cluster partition of the data set.
4  Assign support vectors and inside data points to the lower approximation of their respective cluster based on the
   solution obtained in the labeling phase.
5  Assign bounded support vectors to the fuzzy boundaries of the clusters generated by SVC algorithm.
6  Generate the distance matrix BSV vs. SV to obtain the distance of each data point that is outside of the sphere to each
   support vector.
7  Partition the distance matrix by columns according to the labeling phase.
8  for each cluster do
9      for each x_i ∈ BSV do
10         Get the closest x_j ∈ SV considering those which define the same cluster.
11         Calculate the final membership degree using Eq. 12;
```

Note that steps 1–3 are the Support Vector Clustering algorithm, while steps 4–11 are our proposal. Finally, if the penalty parameter C described in Section 3.1 is set to 1, the set of bounded support vectors will be empty and the fuzzy boundary will not exist because all data points will be classified in a specific cluster becoming elements of the lower approximations of the clusters, and a crisp partition of the data set will be obtained.

3.3. Comprehensible application on example data set

In order to show the main contributions of our algorithm easily, we introduce the Motivation Data Set, an artificially generated data set with 316 instances, 16 of which are located outside the main masses of the two clusters. Fig. 4 displays the raw data of this example.

For this sample data set, we set, by trial and error, the parameters of the Support Vector Clustering algorithm $q = 12$ and $\upsilon = 0.074$. After the training phase, support vectors, bounded support vectors, and inside data points are identified as is shown in Fig. 5, where red points are SV, orange points are BSV, and the remaining ones are inside data.

Then, using the Support Vector Graph labeling rule proposed by Ben-Hur et al. [4], the crisp clusters can be found, and the {0, 1}-membership matrix for the data is obtained. Fig. 6 shows the labeling results. For visualization purposes, only the support vectors have different colors with each color representing a different cluster.

As mentioned in Section 2.2.2, given that images of bounded support vectors are outside the sphere in the higher-dimensional space, well-known labeling rules have left them unclassified with zero membership degree to all the clusters found. In order to overcome this issue, we extend the crisp nature of the current clusters to a rough–fuzzy one, and assign the classified data to the respective lower approximations of the new rough–fuzzy clusters, and the bounded support

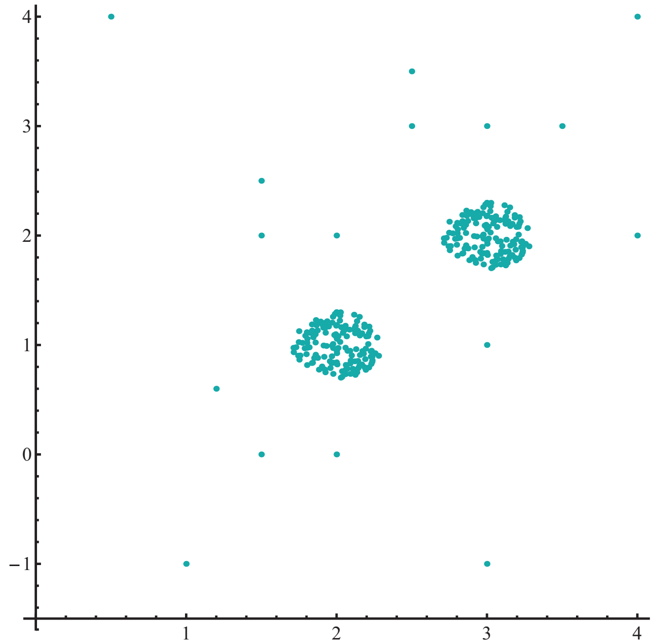


Fig. 4. Motivation data set.

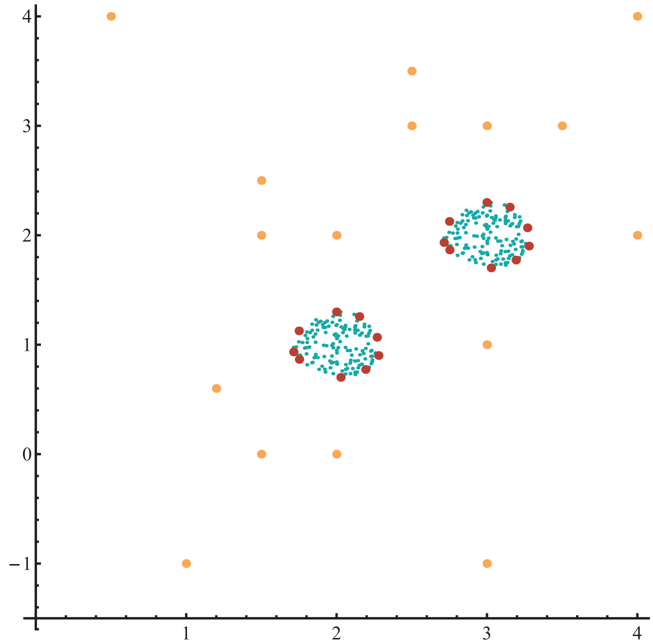


Fig. 5. Results of the training phase of RFSVC algorithm for motivation data set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

vectors to the fuzzy boundaries. After that, we compute the membership degrees of unclassified data using steps 6–11 of Algorithm 1. The results are displayed in Table 3.

From the graphical point of view, nothing changes in the current clusters, however, now we have additional information about bounded support vectors, and with this new knowledge we can detect outliers, given that the membership degrees of BSV are not constrained to sum 1. Looking again at Table 3, if the maximum membership degree for a BSV is very close to zero, we can consider this BSV to be an outlier. For the Motivation Data Set, all BSVs can be considered outliers, but, as we will see in Section 4, this fact is not always true for all data sets.

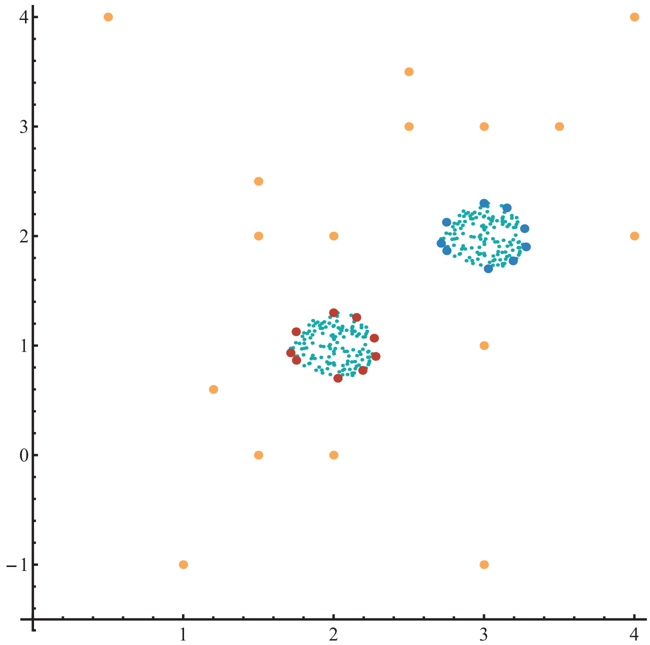


Fig. 6. Results of the labeling phase of RFSVC algorithm for Motivation Data Set.

Table 3
Membership matrix for bounded support vectors

BSV Id	X	Y	$\mu_{i,1}$	$\mu_{i,2}$
301	4	4	9.61657E−58	2.60884E−20
302	2.5	3	4.23234E−17	0.000140085
303	3.5	3	4.91802E−26	0.000309468
304	3	3	2.60884E−20	0.002783015
305	2.5	3.5	2.90258E−27	1.56361E−09
306	4	2	7.18629E−21	0.001749415
307	1.2	0.6	0.011077595	1.25507E−21
308	1.5	2	0.000140085	1.97424E−08
309	2	2	0.002783015	0.00208785
310	1.5	2.5	1.56361E−09	1.34497E−09
311	3	1	0.001749415	0.002697496
312	1.5	0	9.63812E−05	4.90062E−27
313	2	0	0.002697496	2.55448E−21
314	3	-1	1.65352E−20	9.11978E−39
315	1	-1	2.55448E−21	1.60114E−59
316	0.5	4	1.94621E−50	2.08462E−45

4. Experiments and discussion

In this section we introduce the data sets we used to test our procedure first, and explain how we calibrated the initial parameters. Then, in Section 4.2, we present the results obtained using Rough–Fuzzy Support Vector Clustering, Rough–Fuzzy C-Means, and Rough–Possibilistic C-Means. At the same time, we discuss the results, show how our algorithm can detect outliers, and highlight the important advantages of our proposal.

4.1. Description of data sets and experimental set-up

To test our methodology, we used several data sets of different natures, such as benchmark, real world, and artificially generated, with different cluster silhouettes and densities. The main characteristics of the data sets used are summarized in Table 4.

It is necessary to run the Support Vector Clustering algorithm to apply our method for obtaining the set of support vectors, bounded support vectors, and inside data points. The main parameters of SVC are the Gaussian kernel width q which affects the number of support vectors, and the penalty constant $C = \frac{1}{N\upsilon}$ which determines the percentage of data that will be left outside the hyper-sphere in the higher-dimensional space. Many studies have been undertaken in order

Table 4
Data sets characteristics.

Name	Type	Instances	Classes	Attributes
Two Circles	Artificial	4500	2	2
Three Circles	Artificial	6750	3	2
Two Squares	Artificial	4500	2	2
Four Squares	Artificial	9000	4	2
XO	Benchmark	3600	2	2
XOOut	Benchmark	3603	2	2
S1-Gaussian	Benchmark	5000	15	2
Unbalance	Benchmark	6500	8	2
BankNote	Real World	1374	2	4
Glass	Real World	214	6	9
Cancer	Real World	569	2	30
Quake	Real World	2178	NA	4

Table 5
Algorithms' parameters.

Name	RFSVC		RFCM			RPCM			
	q	υ	c	m	w	c	m	w	$\tilde{w}$
Two Circles	5.7	$\frac{1}{3}$	2	2	0.66	2	2	0.66	0.34
Three Circles	5.7	$\frac{1}{3}$	3	2	0.4	3	2	0.6	0.4
Two Squares	5.44	$\frac{1}{3}$	2	2	0.7	2	2	0.7	0.3
Four Squares	8	$\frac{1}{3}$	4	2	0.7	4	2	0.75	0.25
XO	7	$\frac{1}{6}$	2	2	0.45	2	2	0.75	0.25
XOOut	7	$\frac{1}{6}$	2	2	0.45	2	2	0.75	0.25
S1-Gaussian	20	0.4	15	2	NA	15	2	NA	NA
Unbalance	10	0.05	8	2	NA	8	2	NA	NA
BankNote	0.25	0.1	2	2	0.6	2	2	0.6	0.4
Glass	0.1	0.1	6	2	0.7	6	2	0.7	0.3
Cancer	0.1	0.37	2	2	0.6	2	2	0.6	0.4
Quake	1	0.2	3	2	0.6	3	2	0.6	0.4

to calibrate these parameters accurately [35,39]. We calibrated them by simple trial and error because the calibration of parameters would be beyond the scope of this paper. For more details we refer you to [4,35,39].

On the other hand, Rough–Fuzzy C-Means (RFCM) and Rough–Possibilistic C-Means (RPCM) algorithms developed by Maji and Pal [22–24], and Maji and Paul [25] were run in order to compare their results with those of our proposal. The main parameters of these algorithms are the number of clusters c, the fuzzifier m, and the relative weight of lower approximation w.

We set the parameters for Rough–Fuzzy Support Vector Clustering following Ben-Hur's suggestions [4], and similarly, for Rough–Fuzzy C-Means and Rough–Possibilistic C-Means, we used the ideas reported by Maji and Pal [23]. Table 5 shows the parameters set for each one of the methods used in this paper. In some cases, the respective algorithm did not converge, which is indicated by NA.

All data sets used in this paper and the respective results of the algorithms can be downloaded from the following link: https://goo.gl/baeUpt. Benchmark and Real World data sets can also be downloaded from the major well-known data repositories [1,3,32].

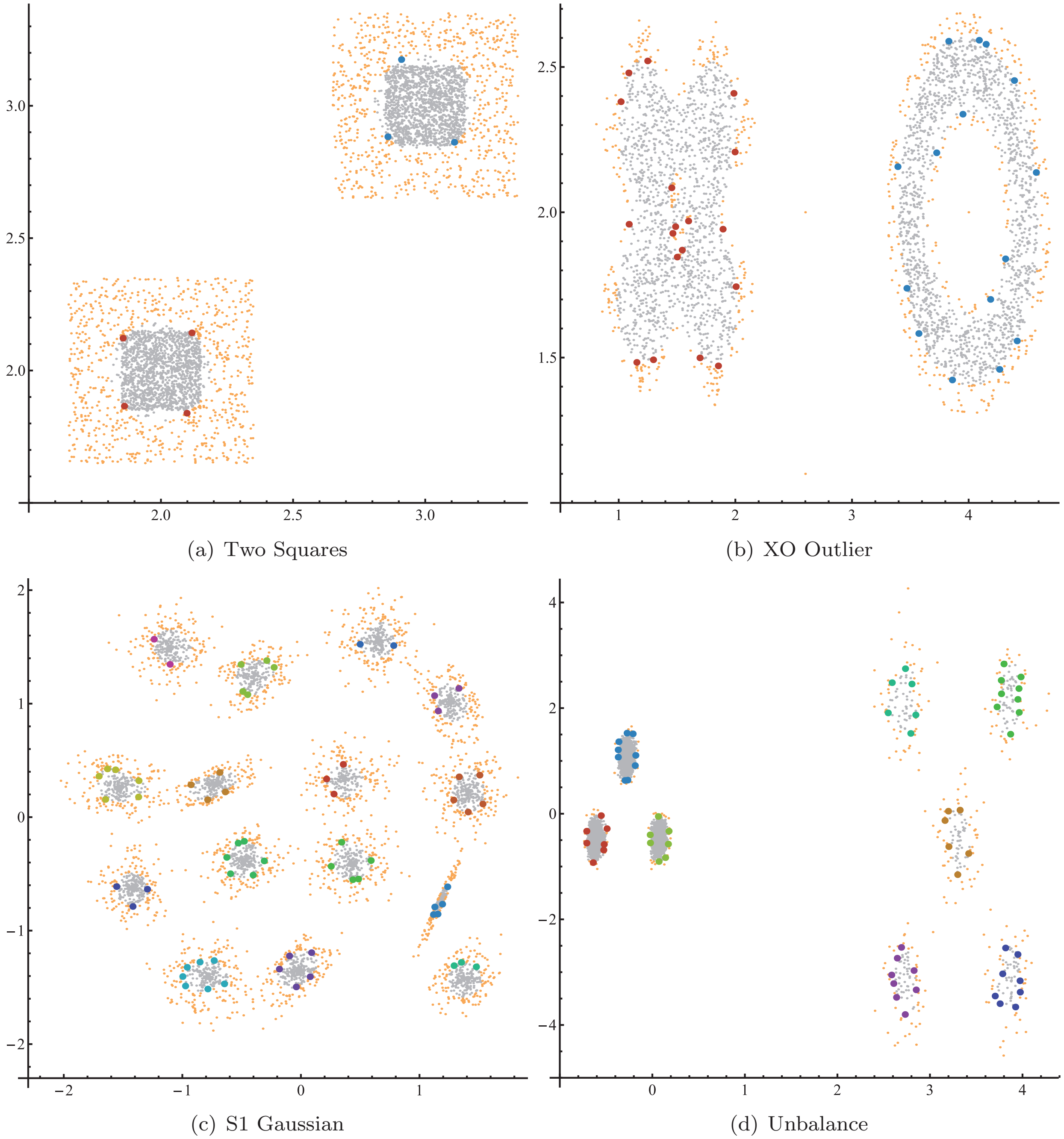
4.2. Results and discussion

In this section, we present first the graphical results of Rough–Fuzzy Support Vector Clustering, Rough–Fuzzy C-Means, and Rough–Possibilistic C-Means for a sample of the two-dimensional data sets tested. Then, quantitative indices proposed by Maji and Pal [23] are computed to compare the results of the algorithms used in this section. Finally, a sample of the membership matrix of the XO Outlier data set is presented to show how our method detects outliers.

For Figs. 7, 8, and 9, orange points belong to fuzzy boundaries, color highlighted points are the prototypes of each cluster, while the remaining gray data points are elements of the lower approximations of their respective classes.

Fig. 7 shows the results graphically for Two Squares, XO Outlier, S1 Gaussian and Unbalance data sets using the RFSVC algorithm. These data sets have different numbers of clusters, shapes and densities, and in all cases the correct number of clusters was found. Additionally, given that support vectors act as cluster prototypes, the shapes of the clusters found can be observed.

Similarly, Fig. 8 shows the results graphically for the same data sets using the RFCM algorithm. In each scenario, the correct number of clusters was provided to the algorithm. In this case, only Two Squares and XO Outlier data sets are



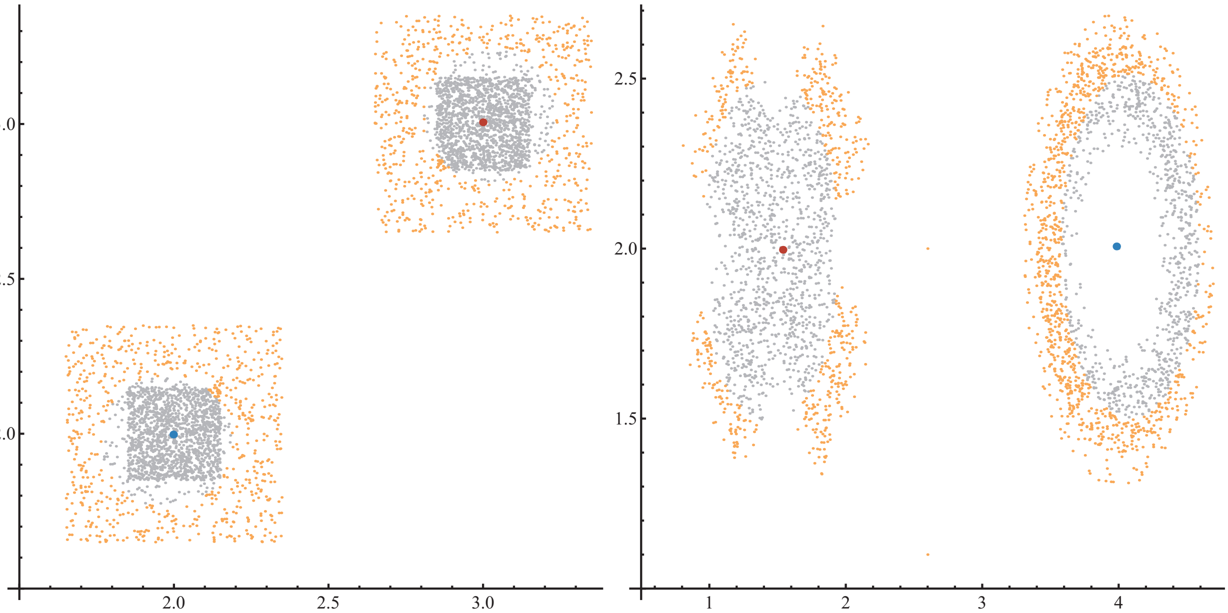
(a) Two Squares (b) XO Outlier (c) S1 Gaussian (d) Unbalance

Fig. 7. RFSVC results for 2D data sets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

displayed, given that the algorithm was unable to converge and find a solution for S1 Gaussian and Unbalance data sets. On the other hand, for the XO Outlier data set, the method returned a data point that was introduced as an outlier in the center of the “O” class as the center of the “O” cluster.

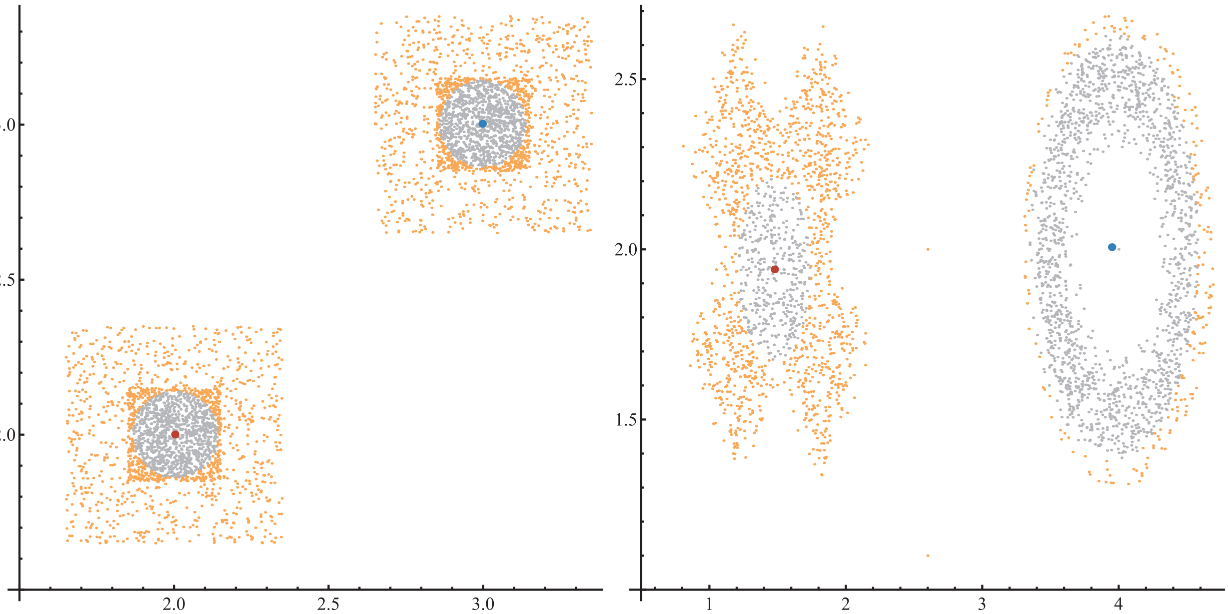
Finally, Fig. 9 shows the results graphically for the same data sets using the RPCM algorithm. Similar to the RFCM, only the Two Squares and XO Outlier data sets are displayed given that the algorithm was unable to converge using S1 Gaussian and Unbalance data sets. Despite this fact, it found perfect circles inside the two squares and the “X” cluster of the XO data set. Furthermore, RPCM assigned the introduced outlier in the center of the “O” cluster to its lower approximation.

In order to evaluate the performance of our algorithm numerically, we calculated Maji’s validity measures [23]. These indices are well suited for this purpose given that they need only the membership matrix and the parameters used by the



(a) Two Squares (b) XO Outlier

Fig. 8. RFCM results for 2D data sets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).



(a) Two Squares (b) XO Outlier

Fig. 9. RPCM results for 2D data sets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

algorithm. Other classical quality indices are usually center-based [12]. As a consequence, they cannot be used to evaluate the performance of our algorithm because it uses the support vectors as prototypes instead of the centers like other approaches.

We briefly present Maji's quantitative indices. In the equations presented below, Lw_j and FB_j are the lower approximation and fuzzy boundary of cluster j, respectively.

Table 6
Validity measures for artificial data sets.

Name	Indice	RFSVC	RFCM	RPCM
Two Circles	α	0.9383	0.8344	***0.9612***
	α^*	0.9382	0.8344	***0.9612***
Three Circles	α	0.9237	0.6329	***0.9500***
	α^*	0.9237	0.6327	***0.9504***
Two Squares	α	0.8968	0.8592	***0.9740***
	α^*	0.8966	0.8592	***0.9740***
Four Squares	α	***0.9427***	0.8694	0.2428
	α^*	***0.9425***	0.8702	0.9046

Table 7
Validity measures for benchmark data sets.

Name	Indice	RFSVC	RFCM	RPCM
XO	α	***0.9785***	0.5379	0.9689
	α^*	0.9787	0.5300	***0.9839***
XO Outlier	α	***0.9802***	0.5390	0.9688
	α^*	0.9803	0.5311	***0.9839***
S1 Gaussian	α	***0.8906***	NA	NA
	α^*	***0.8890***	NA	NA
Unbalance	α	***0.9945***	NA	NA
	α^*	***0.9992***	NA	NA

Table 8
Validity measures for Real World data sets.

Name	Indice	RFSVC	RFCM	RPCM
BankNote	α	***0.9827***	0.7823	0.8174
	α^*	***0.9928***	0.7977	0.9064
Glass	α	***0.9448***	0.8596	0.1629
	α^*	***0.9953***	0.8961	0.8920
Cancer	α	***0.8117***	0.7093	0.4483
	α^*	***0.8117***	0.8086	0.7889
Quake	α	0.7191	***0.8060***	0.3111
	α^*	***0.9677***	0.8138	0.8484

- α *index:* It represents the average accuracy of the c clusters. It is the average of the ratio of the number of objects in the lower approximation to the ratio in the upper approximation of each cluster. It captures the average degree of completeness of knowledge about all clusters. A higher value of α indicates a better cluster solution. This index is given by:

$$\alpha = \frac{1}{c}\left(\sum_{j=1}^{c} \frac{\sum_{x_i \in Lw_j} w(\mu_{i,j})^m}{\sum_{x_i \in Lw_j} w(\mu_{i,j})^m + \sum_{x_i \in FB_j} (1-w)(\mu_{i,j})^m}\right)$$

- ρ *index:* It represents the average roughness of the c clusters and is defined by:

$$\rho = 1 - \alpha$$

  Given this fact, we do not report its values in Tables 6, 7, and 8.
- α^* *index:* It represents the accuracy of the approximation of all the clusters. It captures the exactness of approximate clustering. A higher value of α^* indicates a better cluster solution. This index is given by:

$$\alpha^* = \frac{\sum_{j=1}^{c} \sum_{x_i \in Lw_j} w(\mu_{i,j})^m}{\sum_{j=1}^{c} \left(\sum_{x_i \in Lw_j} w(\mu_{i,j})^m + \sum_{x_i \in FB_j} (1-w)(\mu_{i,j})^m\right)}$$

Returning to the focus of this paper, Tables 6, 7, and 8 present the values of the indices mentioned above using Rough–Fuzzy Support Vector Clustering (RFSVC), Rough–Fuzzy C-Means (RFCM), and Rough–Possibilistic C-Means (RPCM). We highlighted the best values with italic-bold font for each data set tested.

Table 6 presents the quality indices for artificially generated data sets. It can be observed that the RPCM algorithm obtains the best values for the Two Circles, Three Circles, and Two Squares data sets, followed by RFSVC and RFCM. This occurs because these data sets are well-suited for circular center-based algorithms, together with the fact that membership degrees of RPCM are not constrained to sum 1, giving additional flexibility to the method over RFCM which is more sensitive

Table 9
Sample of the membership matrix for XO data set.

Data ID	X	Y	RFSVC		RFCM		RPCM	
			C_X	C_O	C_X	C_O	C_X	C_O
3599	4.2850	2.1713	0.0000	0.5425	0	1	0	1
3600	4.3303	1.9614	0.0000	0.9002	0	1	0	1
3601	4	2	0.0000	0.4426	0	1	0	1
3602	2.6	2	0.0576	0.0104	0.6309	0.3691	0.0417	0.1254
3603	2.6	1.1	0.0077	0.0003	0.5873	0.4127	0.0272	0.0899

to noise. On the other hand, RFSVC outperforms RFCM and RPCM on the Four Squares data set because the shape structure is more complicated, while RPCM finds coincident centers.

Similarly, Table 7 presents quality indices for the benchmark data sets. Contrary to the results of Table 6, our algorithm obtained the best quality indices in all data sets, except for α^*, the index in the two versions of the XO data set, where it gave results only slightly worse than RPCM. However, looking at Figs. 7(b)–9(b), we see that RFSVC achieved a better prototype structure, and assigned outliers correctly to the boundaries of the clusters.

For the Real World data sets, the numbers of clusters found by RFSVC are 2, 2, 1, and 3 for BankNote, Glass, Cancer, and Quake, respectively. For RFCM and RPCM, the correct numbers of clusters were used if they were known in advance, or otherwise, the number found by RFSVC was used.

Table 8 presents validity indices for Real World data sets. Again, the RFSVC algorithm scores the best values in all the data sets except for the Quake data set where RFCM gets a better value for the α index. In these data sets, RPCM leads to coincident centers except for in BankNote where it scores better than RFCM.

A special case occurs with the Glass data set, which has six classes, but RFSVC considers that there are only two clusters. In order to compare RFCM against RPCM we ran both algorithms using $c = 2$ and $c = 6$, leading to better indices in the second case, although they were not sufficient to outperform the results of our algorithm. A possible reason could be that under the unsupervised scenario, two non-ball-shaped clusters fit the data better than the two or six ball-shaped clusters found by RFCM and RPCM.

To conclude this section, Table 9 presents a sample of the membership matrices obtained by applying RFSVC, RFCM, and RPCM algorithms to the XO Outlier data set. This data set is the classic XO data set with three introduced outliers, one of them being the center of the "O" cluster.

As observed, RFCM and RPCM classify the outlier located in the center of the "O" cluster as an element of the lower approximation. Consequently, the membership degree of this outlier is 1. For the remaining outliers, RFCM calculates membership degrees very close to 0.5, making it difficult to recognize them, while RPCM obtains membership degrees close to 0 in both cases, allowing their detection.

Since our approach (RFSVC) is density-based, it recognizes the data point introduced in the center of the "O" cluster as an outlier, and assigns it a membership degree of 0.44 to the fuzzy boundary of the "O" cluster, and $5.02 \times 10^{-17} \cong 0$ to the fuzzy boundary of the "X" cluster. Although the membership degrees in RPCM are not constrained to sum up to 1, this algorithm fails to detect the outlier introduced in the center of the "O" cluster because it uses a center-based approach for obtaining the membership matrix, demonstrating that RPCM does not work effectively in non-spherical clusters.

5. Conclusions and future work

In this paper the novel Rough–Fuzzy Support Vector Clustering algorithm was presented. The numerical and graphical results indicate the potential this approach has in artificial as well as in Real World data sets. In summary, a list of the main contributions of our method is presented.

Soft clusters of any silhouette with lower approximations and fuzzy boundaries are provided together with their respective membership matrices. The membership degrees are calculated using the Gram matrix and support vectors as clusters' prototypes, which is more natural than using a user-defined membership function. The membership degrees of each bounded support vector in all clusters do not necessarily sum 1 as, e.g., in Fuzzy C-Means and Rough–Fuzzy C-Means, so outliers can be detected at the same time. No prior assumption about the number of clusters is necessary. This characteristic is inherited from Ben-Hur's algorithm. Computational time for calculating the [0, 1]-membership matrix is negligible.

Since our method is based on the Support Vector Clustering, it inherits the main advantages of Ben-Hur's algorithm. The data analyst can have the best of both worlds: the crisp partition of the data set using the SVC, and the flexibility of soft computing approaches, by running our algorithm with practically no additional computational cost.

More research has to be done in order to better understand the potential provided by combining SVC with soft computing. Currently, we are working on a dynamic version of RFSVC algorithm and an improved approach for the labeling phase.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions for improving the quality of the paper.

Ramiro Saltos gratefully acknowledges support from CONICYT (CONICYT-PCHA /Doctorado Nacional /2014), SENESCYT (www.senescyt.gob.ec) and the Ph.D. program on Engineering Systems (www.dsi.uchile.cl).

Support for both authors from the Institute on Complex Engineering Systems (www.isci.cl; ICM: P-05-004-F, CONICYT: FBO16) and Fondecyt (1140831) is also gratefully acknowledged.

Appendix A.

Proof of Theorem 1. We first proof the following implication: $d(x_i, x_j) \leq d(x_i, x_k) \Rightarrow d(\phi(x_i), \phi(x_j)) \leq d(\phi(x_i), \phi(x_k))$

$$\begin{aligned}
d(x_i, x_j) &\leq d(x_i, x_k) \\
[d(x_i, x_j)]^2 &\leq [d(x_i, x_k)]^2 \\
\|x_i - x_j\|^2 &\leq \|x_i - x_k\|^2 \\
-q\|x_i - x_j\|^2 &\geq -q\|x_i - x_k\|^2 \\
e^{-q\|x_i - x_j\|^2} &\geq e^{-q\|x_i - x_k\|^2} \\
e^{-q\|x_i - x_j\|^2} - 1 &\geq e^{-q\|x_i - x_k\|^2} - 1 \\
1 - e^{-q\|x_i - x_j\|^2} &\leq 1 - e^{-q\|x_i - x_k\|^2} \\
2(1 - e^{-q\|x_i - x_j\|^2}) &\leq 2(1 - e^{-q\|x_i - x_k\|^2}) \\
\sqrt{2(1 - e^{-q\|x_i - x_j\|^2})} &\leq \sqrt{2(1 - e^{-q\|x_i - x_k\|^2})}
\end{aligned}$$

Given that:

$$\begin{aligned}
d(\phi(x_i), \phi(x_j)) &= \|\phi(x_i) - \phi(x_j)\| \\
&= \sqrt{\phi(x_i)\cdot\phi(x_i) - 2\phi(x_i)\cdot\phi(x_j) + \phi(x_j)\cdot\phi(x_j)} \\
&= \sqrt{2(1 - e^{-q\|x_i - x_j\|^2})}
\end{aligned}$$

$$\therefore d(\phi(x_i), \phi(x_j)) \leq d(\phi(x_i), \phi(x_k))$$

The backward implication could be shown similarly. □

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ins.2015.12.035

References

[1] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data mining software tool: Data set repository, integration of algorithms and experimental analysis framework, J. Mult. Valued Log. Soft Comput. 17 (2-3) (2010) 255–287.

[2] S. Asharaf, S.K. Shevade, M. Narasimba, Rough support vector clustering, Pattern Recognit. 38 (2005) 1779–1783.

[3] M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013 http://archive.ics.uci.edu/ml/citation_policy.html.

[4] A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, J. Mach. Learn. Res. 2 (12) (2001) 125–137.

[5] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algoritms, Plenum Press, 1981.

[6] J.H. Chiang, P.Y. Hao, A new kernel-based fuzzy clustering approach: Support vector clustering with cell growing, IEEE Trans. Fuzzy Syst. 11 (4) (2003) 518–527, doi:10.1109/TFUZZ.2003.814839.

[7] S. Chiu, Extracting fuzzy rules for pattern classification by cluster estimation, in: Proceedings of the Sixth International Conference of Fuzzy Systems Association, 1995, pp. 273–276.

[8] G. Chonghui, L. Fang, An improved algorithm for support vector clustering based on maximum entropy principle and kernel matrix, Expert Syst. Appl. 38 (7) (2011) 8138–8143.

[9] V. D'Orangeville, M.A. Mayers, M.E. Monga, M.S. Wang, Efficient cluster labeling for support vector clustering, IEEE Trans. Knowl. Data Eng. 25 (11) (2013) 2494–2506.

[10] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A Survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (2008) 176–190.

[11] A.L. García, M. González, R.E. Ibarra, J. Mora, N. Hernández, Hybrid fuzzy-SV clustering for heart disease identification, in: Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 2006, p. 121.

[12] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2–3) (2001) 107–145, doi:10.1023/A:1012801612483.

[13] S.S. Keerthi, E.G. Gilbert, Convergence of a generalized SMO algorithm for SVM classifier design, Mach. Learn. 46 (2002) 351–360.

[14] G. Kou, Y. Peng, G. Wang, Evaluation of clustering algorithms for financial risk analysis using MCDM methods, Inf. Sci. 275 (2014) 1–12.

[15] J. Lee, D. Lee, An improved cluster labeling method for support vector clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 461–464.

[16] S.-H. Lee, K.M. Daniels, Cone clustering labeling for support vector clustering, in: Proceedings of the Sixth SIAM International Conference on Data Mining, 2006, pp. 484–488.

[17] S.H. Lee, K.M. Daniels, Gaussian kernel width exploration and cone cluster labeling for support vector clustering, Pattern Anal. Appl. 15 (3) (2012) 327–344, doi:10.1007/s10044-011-0244-8.
[18] H. Li, A fast and stable cluster labeling method for support vector clustering, J. Comput. 8 (12) (2013) 3251–3256.
[19] H. Li, Y. Ping, Recent advances in support vector clustering: Theory and applications, Int. J. Pattern Recognit. Artif. Intell. 29 (01) (2015) 1550002.
[20] P. Lingras, C. West, Interval set clustering of web users with rough *K*-means, J. Intell. Inf. Syst. 23 (1) (2004) 5–16.
[21] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, Inf. Sci. 295 (2015) 407–428.
[22] P. Maji, S. Pal, RFCM: A hybrid clustering algorithm using rough and fuzzy sets, Fundam. Inf. 80 (4) (2007a) 475–496.
[23] P. Maji, S. Pal, Rough set based generalized Fuzzy C-means algorithm and quantitative indices, IEEE Trans. Syst. Man Cybern. Part B 37 (6) (2007) 1529–1540.
[24] P. Maji, S. Pal, Rough-fuzzy clustering: Generalized C-means algorithm, in: Rough-Fuzzy Pattern Recognition: Applications in BioInformatics and Medical Imaging, Jonh Wiley and Sons, Inc., New Jersey, 2012, pp. 47–83.
[25] P. Maji, S. Paul, Rough-fuzzy C-means for clustering microarray gene expression data, in: Perception and Machine Intelligence, Springer, 2012, pp. 203–210.
[26] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (1982) 341–356.
[27] G. Peters, F. Crespo, P. Lingras, R. Weber, Soft clustering: Fuzzy and rough approaches and their extensions and derivatives, Int. J. Approx. Reason. 54 (2013) 307–322.
[28] L. Ping, Z. Chun-Guang, Z. Xu, Improved support vector clustering, Eng. Appl. Artif. Intell. 23 (4) (2010) 552–559.
[29] L. Ping, X. Rong, X. You, M. Xu, Novel three-phase clustering based on support vector technique, J. Softw. 8 (4) (2013) 955–962.
[30] Y. Ping, Y.-J. Tian, Y.-J. Zhou, Y.-X. Yang, Convex decomposition based cluster labeling method for support vector clustering, J. Comput. Sci. Technol. 27 (2) (2012) 428–442.
[31] R. Saltos, R. Weber, Rough-fuzzy support vector domain description for outlier detection, in: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, Istanbul - Turkey, 2015, pp. 1–6, doi:10.1109/FUZZ-IEEE.2015.7337882. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7337882.
[32] Speech and Image Processing Unit, Clustering data sets, 2015. http://cs.joensuu.fi/sipu/datasets/ (accessed 15.12.15).
[33] D. Tax, R. Duin, Support vector domain description, Pattern Recognit. Lett. 20 (11) (1999) 1191–1199.
[34] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, Int. J. Pattern Recognit. Artif. Intell. 25 (3) (2011) 337–372.
[35] J.-S. Wang, J.-C. Chiang, A cluster validity measure with a hybrid parameter search method for the support vector clustering algorithm, Pattern Recognit. 41 (2) (2008) 506–520.
[36] Y. Wang, S. Chen, Soft large margin clustering, Inf. Sci. 232 (2013) 116–129.
[37] R. Xu, D. Wunsch, A survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (03) (2005) 645–678.
[38] J.Y.J. Yang, V.E. Castro, S. Chalup, Support vector clustering through proximity graph modelling, in: Proceedings of the Ninth International Conference on Neural Information Processing (ICONIP), 2, 2002, doi:10.1109/ICONIP.2002.1198191.
[39] O. Yilmaz, L. Achenie, R. Srivastava, Systematic tuning of parameters in support vector clustering, Math. Biosci. 205 (2) (2007) 252–270.
[40] L. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[41] E.-H. Zheng, M. Yang, P. Li, Z.-H. Song, Fuzzy support vector clustering, in: J. Wang, Z. Yi, J. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks, *Lecture Notes in Computer Science*, vol. 3971, Springer Berlin Heidelberg, 2006, pp. 1050–1056.