Technical Report

Scalable Rough Support Vector Clustering

Asharaf S S K Shevade M Narasimha Murty

Computer Science and Automation
Indian Institute of Science, Bangalore - 560012.
{asharaf,shirish,mnm}@csa.iisc.ernet.in

Abstract. In this paper a novel scalable soft support vector clustering algorithm is proposed. Here softness is imparted to Support Vector Clustering paradigm by employing rough set theory and scalability is achieved using Multi Sphere Support Vector Clustering method. Empirical results show that the proposed method gives meaningful cluster abstractions.

1 Introduction

Clustering is a machine learning technique that attempts to find potentially useful but previously unknown groups that may exist in any dataset based on some similarity function. Several algorithms have been proposed in the literature for clustering datasets; viz K-Means[7], CLARA[5], DBSCAN[15], CURE[12], Support Vector Clustering(SVC)[3], etc. These conventional clustering algorithms achieve a hard partitioning of the dataset into crisp sets or groups where a data point is assigned to exactly one cluster. This may not always make sense in real world scenarios where a soft partitioning is more natural. Further, the clusters that exist in any dataset may be of arbitrary shapes. Hence clustering algorithms that we use must have the capability to capture such an abstraction. Soft Clustering approaches like Fuzzy C-Means[16], Rough SOM[2], Rough Clustering method discussed in Voges[17] Rough K-Means[11], ARFL[8] and RSVC[6] are some contributions in this direction. Another major concern in clustering large datasets is the usage of scalable algorithms. Some scalable clustering approaches that exist in the literature are Leader[7], BIRCH[13] and SVC with Cell Growing[4].

This paper introduces a scalable rough set theoretic kernel based method namely Scalable Rough Support Vector Clustering (SRSVC). It achieves Multi Sphere Soft Support Vector Clustering through a natural fusion of rough set theory and the scalable kernel based data clustering method. The organization of this paper is as follows. Section 2 discusses the Support Vector Clustering method. In Section 3 Rough Support Vector Clustering method is given. The Scalable Rough Support Vector Clustering method is introduced in Section 4. Empirical results are given in Section 5 and conclusions given in Section 6.

2 Support Vector Clustering

As in Support Vector Machines here also the computation in a high dimensional feature space is achieved using a Kernel function without explicitly doing the mapping of data points to the high dimensional feature space. In feature space we look for the smallest sphere that encloses the image of the data. If this sphere is mapped back to data space, it forms a set of contours which enclose the data points. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster. The kernel parameters can control the number of clusters. Here the outliers are handled with the help of a soft margin formulation.

To define the formulation, let $\{x_i\} \subseteq X$ be a d-dimensional data set having m points, with $x_i \in R^d$, the data space. Now using a nonlinear transformation ϕ from X to some high dimensional feature space, we look for the smallest sphere of radius R enclosing all the points in X. Now the primal problem[3] can be stated as

$$min\ R^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t\ \ \|\phi(x_i) - \mu\|^2 \leq R^2 + \xi_i, \qquad \xi_i \geq 0\ \forall i \tag{1}$$

Here $C\sum_{i=1}^{m}\xi_i$ is the penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns that lie outside the feature space sphere), μ is the center of the sphere in high dimensional feature space and $\|.\|$ is the L_2 norm.

Since this is a Convex Quadratic Programming problem, it is easy to solve its Wolfe Dual[10] form. The dual formulation is

$$min \sum_{i,j}^{m}\alpha_i\alpha_j K(x_i, x_j) - \sum_{i=1}^{m}\alpha_i K(x_i, x_i)$$

$$s.t\ \ 0 \leq \alpha_i \leq C \quad fori = 1....m, \qquad \sum_{i=1}^{m}\alpha_i = 1 \tag{2}$$

Here $K(x_i, x_j)$ represents the Kernel function giving the dot product $\phi(x_i) \cdot \phi(x_j)$ in the high dimensional feature space and α_is are the Lagrangian multipliers.

The value of α_i decides whether a point $\phi(x_i)$ is inside, on the sphere or outside. The points with $0 < \alpha_i < C$ form the Support Vectors (SVs). Hence the radius of the sphere enclosing the image of the data points is given by

$$R = G(x_i)\ \ where\ \ 0 < \alpha_i < C$$

where

$$G^2(x_i) = \|\phi(x_i) - \mu\|^2 = K(x_i, x_i) - 2\sum_{j=1}^{m}\alpha_j K(x_j, x_i) + \sum_{j,k}^{m}\alpha_j\alpha_k K(x_j, x_k) \quad (3)$$

Now the contours that enclose the points in data space are defined by

$$\{x/G(x) = R\}$$

Thus the computation in high dimensional feature space and reverse mapping to find the contours in data space is avoided with the help of Kernel function. Once these contours are found the cluster assignments are done as follows. This employs a geometric method involving G(x), based on the observation: given a pair of points that belong to different clusters, any path that connects them must exit from the sphere in feature space. So we can define an adjacency matrix M by considering all pairs of points x_i and x_j whose images lie in or on the sphere in the feature space and then looking at the image of the path that connects them as

$$M[i,j] = \begin{cases} 1 \text{ if } G(y) \leq R & \forall y \in [x_i, x_j] \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

Clusters are now defined as the connected components of the graph induced by M. The points that lie outside the sphere (Bounded Support Vectors) can be assigned to the closest clusters.

3 Rough Support Vector Clustering

Rough Support Vector Clustering(RSVC) is an extension of Support Vector Clustering paradigm employing rough set theory to achieve soft clustering. To discuss the method formally, let us use the notion of rough sets to define a *Rough Sphere*. A *Rough Sphere* is defined as a sphere having an inner radius R defining its lower approximation and an outer radius $T > R$ defining its upper approximation. As in SVC, Rough SVC also uses a kernel function to achieve computation in a high dimensional feature space. It tries to find the smallest Rough Sphere in the high dimensional feature space enclosing the images of all the points in the dataset. Now those points whose images lie within the Lower Approximation(L_A) are points that definitely belong to exactly one cluster (hard core of a cluster) and those points whose images lie in the Boundary Region(B_R) (in the upper approximation but not in lower approximation) may be shared by more than one cluster (soft core of the clusters). Some points are permitted to lie outside the sphere and are termed outliers. By using a nonlinear transformation ϕ from data space to some high dimensional feature space, we look for the smallest enclosing rough sphere of inner radius R and outer radius T. Now the primal problem can be stated formally as

$$min \quad R^2 + T^2 + \frac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i + \frac{\delta}{\upsilon m}\sum_{i=1}^{m}\xi_i^{'}$$

$$s.t \quad \|\phi(x_i) - \mu\|^2 \leq R^2 + \xi_i + \xi_i^{'} \qquad 0 \leq \xi_i \leq T^2 - R^2 \qquad \xi_i^{'} \geq 0 \quad \forall i \qquad (5)$$

Here $\frac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i$ is the penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns falling in the boundary region) and $\frac{\delta}{\upsilon m}\sum_{i=1}^{m}\xi^{'}$ is the penalty term associated with the patterns whose distance from the center of the sphere in feature space is greater than T (patterns falling outside the rough sphere).

Since this is a Convex Quadratic Programming problem it is easy to write its Wolfe Dual[10]. The Lagrangian can be written as

$$L = R^2 + T^2 + \frac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i + \frac{\delta}{\upsilon m}\sum_{i=1}^{m}\xi_i^{'} + \sum_{i=1}^{m}\alpha_i(\|\phi(x_i) - \mu\|^2 - R^2 - \xi_i - \xi_i^{'}) -$$

$$\sum_{i=1}^{m}\beta_i\xi_i + \sum_{i=1}^{m}\lambda_i(\xi_i - T^2 + R^2) - \sum_{i=1}^{m}\eta_i\xi_i^{'}$$

$$\textit{where the Lagrange multipliers} \quad \alpha_i \geq 0 \qquad \beta_i \geq 0 \qquad \lambda_i \geq 0 \qquad \eta_i \geq 0 \quad \forall i \quad (6)$$

Using Karush-Kuhn-Tucker(KKT) conditions on equation(6) we get

$$\sum_{i=1}^{m}\alpha_i = 2 \qquad \mu = \frac{1}{2}\sum_{i=1}^{m}\alpha_i\phi(x_i)$$

$$\beta_i - \lambda_i = \frac{1}{\upsilon m} - \alpha_i \qquad \frac{\delta}{\upsilon m} - \alpha_i = \eta_i$$

$$\alpha_i(\|\phi(x_i) - \mu\|^2 - R^2 - \xi_i - \xi_i^{'}) = 0$$

$$\lambda_i(\xi_i - T^2 + R^2) = 0$$

$$\beta_i\xi_i = 0 \qquad \eta_i\xi_i^{'} = 0$$

From the above equations the Wolfe Dual[10] form can be written as

$$min \sum_{i,j}^{m} \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i K(x_i, x_i)$$

$$s.t \quad 0 \leq \alpha_i \leq \frac{\delta}{\upsilon m} \quad for\ i = 1....m, \qquad \sum_{i=1}^{m} \alpha_i = 2 \tag{7}$$

Here $K(x_i, x_j)$ represents the Kernel function giving the dot product $\phi(x_i).\phi(x_j)$ in the high dimensional feature space[3].

For RSVC, $\delta > 1$ and it reduces to the original SVC formulation for $\delta = 1$. Also the values of α_i decide whether the pattern x_i falls in the lower approximation or the boundary region or outside the feature space rough sphere. From KKT conditions on equation(6), it can be observed that image of points with:

- $\alpha_i = 0$ lie in lower approximation.
- $0 < \alpha_i < \frac{1}{\upsilon m}$ form the Hard Support Vectors (Support Vectors which mark the boundary of lower approximation).
- $\alpha_i = \frac{1}{\upsilon m}$ lie in the boundary region (patterns that may be shared by more than one cluster).
- $\frac{1}{\upsilon m} < \alpha_i < \frac{\delta}{\upsilon m}$ form the Soft Support Vectors (Support Vectors which mark the boundary of upper approximation).
- $\alpha_i = \frac{\delta}{\upsilon m}$ lie outside the sphere (Bounded Support Vectors).

3.1 Cluster Assignment

Once the dual problem is solved to find α_i values, the clusters can be obtained using the following strategy. Let us define

$$R = G(x_i) \quad : \quad 0 < \alpha_i < \frac{1}{\upsilon m}$$

$$T = G(x_i) \quad : \quad \frac{1}{\upsilon m} < \alpha_i < \frac{\delta}{\upsilon m}$$

where $G(x_i)$ is defined in equation(3).

From the above equations we can define the contours that enclose the lower approximation of clusters in data space as

$$\{x / G(x) = R\}$$

and the contours that enclose the upper approximation of clusters in data space as

$$\{x / G(x) = T\}$$

Now the soft clusters in data space are found using a strategy similar to the one used in Support Vector Clustering. This algorithm can be given as

algorithm find_clusters

{

- As in SVC find the adjacency matrix M by considering all pairs of points x_i and x_j whose images in feature space either belong to the Lower Approximation of the rough sphere or are Hard Support Vectors and then looking at the image of the path that connects them as

$$M[i,j] = \begin{cases} 1 \text{ if } G(y) \leq R & \forall y \in [x_i, x_j] \\ 0 \text{ otherwise} \end{cases}$$

- Find connected components for the graph represented by M. Each connected component found gives the Lower Approximation of a cluster C_i.
- Now find the Boundary Regions as
  $x_i \in L\_A(C_i)$ and pattern $x_k \notin L\_A(C_j)$ for any cluster j,
  if $G(y) \leq T \ \ \forall y \in [x_i, x_k]$ then $x_k \in B\_R(C_i)$

}

3.2 Role of δ and υ

From equation(7) we can see that the number of Bounded Support Vectors $n_{bsv} < 2\frac{\upsilon m}{\delta}$. For $\delta = 1$, $n_{bsv} < 2\upsilon m = \upsilon^{'} m$ where $\upsilon^{'} = 2\upsilon$. This corresponds to all the patterns x_i with $\|\phi(x_i) - \mu\|^2 > R^2$. Since $\delta > 1$ for RSVC, we can say that $\frac{\upsilon^{'}}{\delta}$ is the upper bound on the fraction of points permitted to lie outside T and $\upsilon^{'}$ is the upper bound on the fraction of points permitted to lie outside R. Hence υ and δ together give us control over the width of boundary region and the number of Bounded Support Vectors. Hence we can choose the values of υ and δ based on how much percentage of the data we want to put in the soft core of the clusters and how much percentage we want to treat as outliers.

4 Scalable Rough SVC

This is a scalable soft clustering method employing the Multi Sphere Support Vector Clustering[4] approach. The algorithm achieves soft clustering by using the RSVC approach discussed in the last section. Here each cluster is computed using a separate Rough Sphere enclosing the images of all the points of that cluster in the high dimensional feature space. There are three processes involved in SRSVC.

Winning Cluster Competition Process:

For each input data point x, the nearest candidate cluster is considered as the winning cluster. There are two possible ways to compute the distance viz; **a**) the distance $G(x)$ between x and spherical center of the cluster C_k in feature space as given by equation(3) and **b**) the distance between x and cluster C_k in data space as

$$d(x, C_k) = \left(\frac{1}{\mid C_k \mid} \sum_{x_j \in C_k} \|x - x_j\|^2 \right)^{1/2} \tag{8}$$

where $\mid . \mid$ represent the cardinality.

Validity Test:

The vigilance function defines the vigilance degree that x belongs to cluster J, denoted as $g_J(x)$. The data point belongs to winning cluster J if $g_J(x) \leq \epsilon$, the vigilance threshold. Two major factors were considered to decide the vigilance degree viz; **a**) the distance between point x and cluster C_J in data space denoted by $d(x, C_J)$ and **b**) the distance $(G_J(x) - R_J)$ in feature space where $G_J(x)$ is the distance of x from the center of the rough sphere enclosing cluster C_J and R_J is the inner radius of rough sphere enclosing cluster C_J. Both these distances are first normalized as follows

$$d1 = \frac{1}{1 + exp(-p_1 \times [d(x, C_J)]^2)}$$

$$d2 = \frac{1}{1 + exp(-p_2 \times [max(0, (G_J(x) - R_J))]^2)} \tag{9}$$

where p_1 and p_2 are scaling parameters which can be chosen based on the weights we want to give to the distance $d(x, C_J)$ computed in data space and the distance $(G_J(x) - R_J)$ computed in feature space respectively in the computation of vigilance degree $g_J(x)$. A high value of p_1 or p_2 compared to the other result in more weight being assigned to the corresponding distance value in the computation of vigilance degree.

Now the vigilance degree $g_J(x)$ that x belongs to cluster J is given by

$$g_J(x) = \begin{cases} 0 & \text{if } G_J(x) < R_J \\ \sum_{j=1}^{2} w_j b_j & \text{otherwise} \end{cases} \tag{10}$$

where b_j is the jth largest of d_i and $w_j = f(j/2) - f((j-1)/2)$ for an $f(\cdot)$ with $f(x) \geq f(y)$ *for* $x > y$, $f(0) = 0$ and $f(1) = 1$. So $\sum_{j=1}^{2} w_j = 1$ and $w_j > 0 \quad \forall j$.

Learning Process:

If the vigilance degree that data point x belongs to winning cluster J, $g_J(x) \leq \epsilon$ then the algorithm enters the parameter learning process for cluster J to include x in J by performing RSVC.
Now the SRSVC algorithm can be given as

4.1 Variables and Parameters

The variables used by the algorithm are
C_k : Set of data points in cluster k.
HSV_k : Hard Support Vectors for cluster k, $HSV_k \subseteq C_k$.
SSV_k : Soft Support Vectors for cluster k, $SSV_k \subseteq C_k$.
n_c : Current number of Clusters.
R_k : Inner Radius of Rough sphere k in feature space.
T_k : Outer Radius of Rough sphere k in feature space.
J : Index for winning cluster.

The user given parameters of the algorithm are
δ and υ : Cluster parameters. ϵ : Vigilance Threshold.

4.2 The Algorithm

1. **Initialization:** `For first Cluster set`
   $C_1 = HSV_1 = SSV_1 = \{x_1\}, \ R_1 = T_1 = 0, \ \alpha_1 = 1$
2. **Input Points Presentation:**
   `Perform the following computation for` $x_i, \quad i = 2, ..., m$
3. **Winning Cluster Competition:**
   `Find winning cluster` J `such that`

$$d(x_i, C_J) = \min_k \{d(x_i, C_k)\} \quad where \quad k = 1, ...n_c$$

4. **Validity Test:**
   IF $g_J(x_i) <= \epsilon$, `THEN` x_i `belongs to cluster` J, `goto Step 5`
   ELSE `If` $G(x_i) < T_J$ `then add` x_i `to all those clusters` K `with` $G(x_i) < T_K$ `and go to step 5. Else goto step 6`
5. **Parameter Learning:**
   `Perform RSVC for the cluster/clusters where the data point got added and update the cluster parameters. Goto Step 2.`
6. **Create a New Cluster :** `Create a new cluster as`
   $C_{n_c} = HSV_{n_c} = SSV_{n_c} = \{x_i\}, \ R_{n_c} = 0, \ T_{n_c} = 0,$
   `Set the corresponding` α_i `to 1. Return to Step 2`

5 Experimental Results

Experiments are done with a synthetic dataset and a real world Optical Character Recognition dataset; viz Optical Recognition of Handwritten Digits(Optidigits) dataset from the UCI Machine Learning archive available at "*http://www.ics.uci.edu/~mlearn/MLSummary.html*". The clustering results obtained for the Synthetic dataset using SRSVC are given in Figure 1. The Single Pass K-Means(SPKM) algorithm[9], Multi Sphere SVC(MSVC)[4] algorithm, SRSVC algorithm are implemented and a comparative study is made using optidigits dataset. Here the clustering algorithms are used for prototype selection. In all the experiments Gaussian kernel[3] given by $K(x_i, x_j) = e^{-q\|x_i - x_j\|^2}$ is used in MSVC and SRSVC. Here q is a user given parameter.

The Optidigits dataset has 3823 patterns in the Training set and 1797 patterns in the Test set belonging to ten different classes 0..9. Each pattern has 64 attributes. The results on this dataset using different clustering algorithms are given Table 1. Here Nearest Neighbour Classifier(NNC)[7] in the data space with Euclidean distance as the dissimilarity metric is used as the classifier. Table 1 shows the Number Of Prototypes(NOP) selected from the training set by each algorithm and the percentage Classification Accuracy(CA) obtained on the test set using the prototypes as training patterns for NNC.

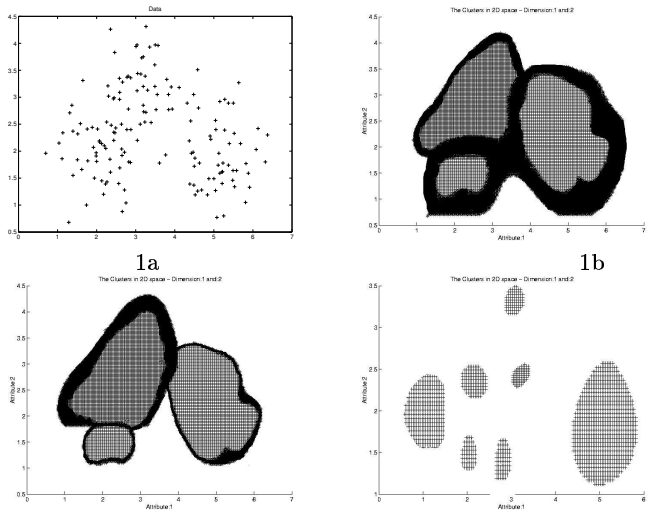

Fig. 1. Clusters obtained for Synthetic Data. *Fig* 1*a* : *Data*. *Fig* 1*b* : $\upsilon = 0.15, \delta =$ 1.2,q= 2.5 *and* $\epsilon = 0.72$ *Fig* 1*c* : $\upsilon = 0.15, \delta = 1.1$,q= 2.5 *and* $\epsilon = 0.72$ *Fig* 1*d* : $\upsilon = 0.25, \delta = 1$,q= 2.8 *and* $\epsilon = 0.69$. Here the light shaded regions in Fig 1b and Fig 1c show the hard core and the hard shaded regions show the soft core of clusters. In Fig 1d the clusters found are identical to those found by SVC and there are no soft core regions for clusters

Algorithm					
SPKM		MSVC		SRSVC	
NOP	CA	NOP	CA	NOP	CA
2581	96.434	2639	95.656	2581	97.032
3182	96.545	3251	97.151	3182	97.426
3377	96.490	3460	97.534	3377	97.758
3670	96.603	3706	97.703	3670	97.985
3742	96.603	3711	97.704	3742	97.985

Table 1. Results on the Optidigits Dataset

6 Conclusion

Here a novel scalable soft support vector clustering approach is introduced. This method employs the Multi Sphere Support Vector Clustering strategy to divide the complex quadratic optimization problem encountered while handling large data sets into smaller subproblems. Hence it is scalable. Some other merits of the proposed method are

- Soft clustering enables us to identify ambiguous regions (having overlap between clusters of heterogeneous nature) in the dataset.
- Soft clusters of any arbitrary shape will be found by the algorithm.
- The user defined parameters allow to control the extent of softness by controlling the width of boundary region and number of outliers.
- Since each region in the partition obtained using this method refines the apriori information about the class labels, this can be extended to detect outliers.

References

1. Zdzislaw Pawlak. Rough Sets, International Journal of Computer and Information Sciences, 11,1982. pp. 341-356.
2. Sankar K Pal and Pabitra Mitra. Pattern Recognition Algorithms for Data Mining: Scalability, Knowledge Discovery and Soft Granular Computing. Chapman & Hall CRC Press, Boca Raton, FL. May 2004.
3. Asa Ben-Hur, David Horn, Hava T. Siegelmann and Vladimir Vapnik. Support Vector Clustering, Journal of Machine Learning Research, 2 (2), 2001. pp. 125-137.
4. Chiang Jung-Hsien and Pei-Yi Hao. A New kernel-Based Fuzzy Clustering Approach: Support Vector Clustering with Cell Growing, IEEE Trans. on Fuzzy Systems, 11, 2003. pp. 518-527.
5. Raymond T Ng, Jiawei Han. Efficient and Effective Clustering Methods for Spacial Data Mining, In Proc. of the VLDB Conference, Chile. 1994.
6. S. Asharaf, S K Shevade and M. N. Murty. Rough Support Vector Clustering, Pattern Recognition. 38(10), 2005. pp. 1779-1783.

7. Sushmita Mitra, Tinku Acharya. Data Mining: Multimedia, Soft Computing, and Bioinformatics, John Wiley & Sons Inc. September 2003.
8. S. Asharaf and M. N. Murty. An Adaptive Rough Fuzzy Single Pass Algorithm for Clustering Large Data Sets, Pattern Recognition, 36(12), 2003, pp. 3015-3018.
9. Fredrik Farnstrom, James Lewis, Charles Elkan. Scalability for Clustering Algorithms Revisited, SIGKDD Explorations, 2(1), 2000. pp. 51-57.
10. Fletcher. Practical Methods of Optimization, 2nd ed., Wiley-Interscience, New York, August 2000.
11. Lingras, P.J. and West C. Interval Set Clustering of Web Users with Rough K-means, Journal of Intelligent Information System, 23(1), 2004. pp. 5-16.
12. Guha S, Rastogi R, Shim K. CURE : An Efficient Algorithm for Clustering Large Databases, Proc. of ACM SIGMOD International Conference on Management of Data, Seattle, Washington, 1998.
13. Tian Zhang, Raghu Ramakrishnan, Miron Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases, Proc. ACM SIGMOD Int. Conference on Management of Data, Montreal, Canada, ACM Press, New York, 1996. pp. 103-114.
14. Spath H. Cluster Analysis Algorithms for Data Reduction and Classification of Objects, Ellis Horwood Limited, West Sussex, U K, 1980.
15. M. Ester, H.P Kriegel, J Sander and X. Xu. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In Proc. of the second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, August 1996. pp. 226-231.
16. Raghu Krishnapuram, Anupam Joshi, Olfa Nasraoui, Liyu Yi. Low-Complexity Fuzzy Relational Clustering for Web Mining, TFS, 9(4), 2001. pp. 595–607.
17. K. E Voges, Research techniques derived from rough sets theory: rough classification and rough clustering, In Proc. of the 4th European Conference on Research Methodology for Business and Management Studies, Paris, April 2005. pp. 437-444.