# Constrained K-Means Classification

Panagiotis N. Smyrlis
Department of Informatics and
Telecommunications Engineering
University of Western Macedonia
Kozani, Greece
pan.smyrlis@gmail.com

Dimosthenis C. Tsouros
Department of Informatics and
Telecommunications Engineering
University of Western Macedonia
Kozani, Greece
dtsouros@uowm.gr

Markos G. Tsipouras
Department of Informatics and
Telecommunications Engineering
University of Western Macedonia
Kozani, Greece
mtsipouras@uowm.gr

*Abstract*—**Classification-via-clustering (CvC) is a widely used method, using a clustering procedure to perform classification tasks. In this paper, a novel K-Means-based CvC algorithm is presented, analysed and evaluated. Two additional techniques are employed to reduce the effects of the limitations of K-Means. A hypercube of constraints is defined for each centroid and weights are acquired for each attribute of each class, for the use of a weighted Euclidean distance as a similarity criterion in the clustering procedure. Experiments are made with 42 well–known classification datasets. The experimental results demonstrate that the proposed algorithm outperforms CvC with simple K-Means.**

***Keywords-classification-via-clustering; k-means; supervised learning***

## I. INTRODUCTION

Data science and especially data mining [1] is a rapidly evolving field with the extraction of valuable knowledge out of large accumulated information being a major challenge. Recent technological progress has led to the creation of large datasets to all scientific areas. Thus, developing methodologies which discover knowledge out of raw data is now many researchers' common concern. Supervised learning methods are an important part of machine learning and data mining, referring to the task of learning from labeled training data, with classification being the most common. Many sophisticated classification algorithms have been proposed in the literature, each exploiting the data in a different way, such as Artificial Neural Networks [2], Bayesian Classifiers [3], Rule Based Classifiers [4, 5], Decision Trees [6, 7] and Ensemble Classifiers [8]. A widely used method is Classification-via-clustering (CvC) [9-14]. The latter involves a clustering technique, in order to group data into clusters, mapped to the classes. Instances are assigned to the clusters or classes according to some criterion (depending on the clustering algorithm used). The most known algorithm in clustering is K-Means [15, 16], which is computationally light. Its low complexity makes it suitable to use in big data analysis for clustering and classification. It is referred to a clustering procedure which pattern for each cluster is a centroid. As a similarity criterion, a sort of distance is used (most commonly the squared Euclidean). According to that criterion, each data is assigned to the proper cluster and the centroids are then recalculated given the updated cluster. This algorithm is widely used in CVC, where the clusters are matched to classes. COP-KMeans [17] is a variation of K-Means, using background knowledge in terms of data-level constraints. This includes two kinds of constraints, must-link and cannot-link, defining whether a data must or must not be in a cluster, neither of which can be violated in order to assign a data point to a cluster. So, if a cluster includes a data-point which must be in the same cluster with another, the latter must be assigned to that cluster. Also, if two data-points are parts of a cannot-link constraint, they cannot be in the same cluster. In the same manner, another technique proposes an extended set of the above set of constraints [18]. Must-link and cannot-link constraints remain, and two more are examined. The latter are based on calculating the maximum distance that two data-points in the same cluster and the minimum distance that two data-points in different clusters must have.

Another approach is Constrained K-Means [19], which includes cluster-level constraints; each cluster has a constraint which is a threshold of the minimum number of data to contain. So, the final centroids are specified when all the clusters contain the desired amount of data. A different method of finding the k centroids is Global K-Means [20], where the problem is solved incrementally. An iterative procedure begins solving the problem initially for one centroid and each iteration adds one centroid to the solution until the k centroids are found. “Fast Global K-Means” [20] tries to accelerate the above procedure. Instead of solving the k-centroids' problem given the k-1 solution, tries to calculate the error reduction from inserting the new centroid. The calculated error is used as a threshold on the next execution of k-means. A different approach is “Fuzzy C-Means” algorithm [21], with each datum fractionally belonging to all clusters and contributing to all centroids' update. In this work, a K-Means based CvC algorithm is presented, introducing the use of constraints for the centroids movement and a weighted Euclidean distance as a similarity criterion. The proposed algorithm is evaluated with 42 well known datasets from the UCI Machine Learning Repository [22]. The 10-fold cross-validation [23] is employed and classification accuracy, average sensitivity and average precision [24] are used as metrics of interest. Experimental results demonstrate that the proposed algorithm considerably outperforms CvC with simple K-Means.

## II. CVC WITH SIMPLE K-MEANS

Clustering is a method of finding subgroups within observations. The most known clustering algorithm used for

CvC is K-Means, which divides the $N$ instances of a dataset into $k$ clusters. Since the number of clusters to be extracted is predefined, K-Means can be easily used in classification by setting $K$ to be equal or greater than the number of classes and mapping each cluster to a specific class. In K-Means based CvC, $K$ is usually set equal to the number of classes $C$. K-Means forms the clusters using the training set. Each cluster is mapped to the appropriate class using the labels of the training data. The majority class of the instances of each cluster is assigned to it. In the classification process, the unlabeled instances are assigned to the closest centroid of the clusters formed in the previous step. The formulation of the clusters using K-Means algorithm is summarized in the following simple K-Means algorithm:

```
Input: $D_T$ ($D_T$: the training set)
Output: $m$ ($m$: the means (centroids))
    1: $t$=0;
    2: initialize randomly the C means $m_1^t$,…,$m_C^t$;
    3: do
    4:        $C$=AssignInstances($D_T$, $m^t$);
    5:        $m^{t+1}$=UpdateMeans($C$);
    6:        $t$++;
    7: while $m^t$!=$m^{t-1}$
    8: return $m^t$;
```

K-Means consists of three main steps. At first, the $k$ means are initialized randomly (line 2). Next, the algorithm assigns the training instances to the "nearest"' mean's cluster (line 4) and updates the means to the centroid of the formulated cluster (line 5). To assign each instance $x_i$, $i = 1, \ldots, N$, to a cluster, the squared Euclidean distance $d_c$, $c = 1, \ldots, C$, from each mean is calculated, as shown in (1).

$$d_c = \sum_{a=1}^{A} (x_{a,i} - m_{a,c})^2 \qquad (1)$$

where $A$ is the number of attributes. After the calculation of the squared Euclidean distance, the instance is assigned to the cluster $CL_c$ where $d_c \leq d_j$ for every $j \in 1, \ldots, C$. The third step of k-means is the recalculation of each mean to be the centroid of all the instances currently in the cluster:

$$m_{a,c} = (\sum_{x_i \in CL_c} x_{a,i})/|CL_c| , \qquad (2)$$

$|CL_c|$ is the size of the cluster, in other words the number of instances that are assigned to it. In K-Means, if $k$ (the number of clusters) and $A$ (the number of attributes) are fixed, the problem can be exactly solved in time $O(N^{Ak+1})$, where $N$ is the number of instances to be clustered [25]. The K-Means algorithm has some limitations and weaknesses [26 - 28] that influence negatively its classification results. First of all, the random initialization of the centroids affects significantly the formation of the cluster-classses. Furthermore, due to clustering task small classes may be dominated by bigger classes, so the grouping may demarcate the bigger class instances into 2 separate classes and include the instances of the small class to one of the bigger clusters. Moreover, the presence of noise - outliers in the dataset can affect the cluster pulling its centroid to a 'bad' position. Another factor that should be taken into account is the distance criterion for the assignation of each instance to a centroid. In CvC, each class may have a difference standard deviation in each attribute than the other classes, thus each attribute may have different contribution in the classification.

## III. Constrained K-Means Classification Algorithm

Constrained K-Means Classification (C-K-Means), a novel CvC algorithm, is proposed in this work. The algorithm is designed so as to address the limitations and weaknesses mentioned above. As classification is a supervised learning technique, background knowledge is used efficiently to extract the model. Two main alterations to K-Means are proposed to use background knowledge: (i) application of constraints to the initialization and update of the centoids, (ii) weighted euclidean distance fuction employment. The training data are used to acquire constraints for each centroid and the weights for each attribute and class. Since each hypercube of constraints is generated based on data from a single class, each centroid inherits the class of the respective hypercube. The clustering procedure and the formulation of the clusters takes place in the test data rather than the training set. This helps us to better classify the testing instances, not only using the distance from each observation from the centroids, but also updating the centroids to the test set clusters formation.

### A. Description of C-K-Means

The model produced during the training procedure includes a hypercube of each class and the weights for each attribute and each class. The algorithm takes as input a training dataset with $A$ attributes and $C$ classes and a hypercube is defined for each class by calculating minimum and maximum bounds for each attribute:

$$b_{a,c}^{min} = avg_{a,c} - l * s_{a,c} \qquad (3)$$

$$b_{a,c}^{max} = avg_{a,c} + l * s_{a,c} \qquad (4)$$

with $avg_{a,c}$ being the average value of the $a$ attribute ($a \in [1, A]$) of the instances belonging to the $c$ class ($c \in [1, C]$), and $s_{a,c}$ the respective standard deviation. The minimum and maximum bounds are $b_{a,c}^{min}$ and $b_{a,c}^{max}$, respectively and $l$ is a parameter defining the relaxation of these bounds. The bounds are used to limit the movement of the centroids and so the cluster formation. The weights are used to facilitate a weighted Euclidean distance calculation as a similarity criterion to the clusters during clustering procedure. The concept includes the estimation of the standard deviation of each class' features in order to define an importance factor for that to be used in the similarity criterion. The calculation of the weights is given by the following formula:

$$w_{a,c} = 1 - \frac{s_{a,c}}{max_{a,c} - min_{a,c}} \qquad (5)$$

with $w_{a,c}$ being the calculated weight, $s_{a,c}$ being the standard deviation of the $a$ attribute ($a \in [1, A]$) of the observations belonging to the $c$ class ($c \in [1, C]$), and $max_{a,c}$, $min_{a,c}$ the respective maximum and minimum values. The weights are then normalized:

$$w_{a,c} = \frac{w_{a,c}}{\sum_{i=1}^{C} w_{a,i}} \qquad (6)$$

After the acquisition of the hypercube of constraints and the weights, the algorithm is ready for the CvC of the unlabeled instances. K-means algorithm is applied as follows:

*Initialization step*: The initial $K$ centroids are generated randomly within the hypercubes, with one centroid in each hypercube ($k = C$). Thus:

$$b_{a,c}^{min} \leq m_{a,c} \leq b_{a,c}^{max} \qquad (7)$$

where $m_{a,c}$ is the centroid value of class $c$ in attribute $a$.

*Assignment step*: In this step, the weighted Euclidean distance is employed as similarity criterion, with each observation assigned to the closest centroid. Weighted Euclidean distance is calculated as follows:

$$d_c = \sqrt{\sum_{a=1}^{A} w_{a,c} * (x_{a,i} - m_{a,c})^2} \qquad (8)$$

*Update step:* If during this procedure, a centroid is to be positioned outside the class hypercube of constraints (in which is initilly created), then it is forced to the bound values, i.e. if the value is less than its class $b_{a,c}^{min}$, then it is set to $b_{a,c}^{min}$, and the same applies if it is greater than the respective $b_{a,c}^{max}$.

After the convergence of the clustering procedure, each instance is classified to the class of the cluster of the centroid that it was assigned in the last iteration.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

To set up the experimental procedure, the discussed algorithm is implemented using C++. The relaxation parameter $l$ in (3) and (4), is experimentally set to 0.1. Also, all attributes of the datasets are normalized. In order to evaluate the proposed algorithm, a comparative study with CvC with simple K-Means took place and classification accuracy, average sensitivity and average precision performance metrics are extracted. For that purpose, WEKA software [29] was employed, where the respective results for CvC with Simple K-Means were obtained. The number of clusters is set equal to the number of classes of each dataset. Both algorithms are tested with 42 datasets from the UCI Machine Learning Repository, with various characteristics. The datasets, number of attributes and instances are presented in Table I (the number of attributes does not include the class). Classification accuracy, average sensitivity and average precision are obtained for each dataset; all results are shown in Table II. The differences per dataset are illustrated in Figures 1, 2 and 3 for the classification accuracy, average sensitivity and average precision, respectively. The diagrams were created as follows: the result obtained with simple K-Means was subtracted by the respective C-K-Means result (for each dataset) and then the results were sorted in a descending order.

The experimental study was based on an extremely large selection of classification datasets (42), including all classification datasets reported as “Most Popular” in the website, and several other well-known datasets. The main exclusion criterion was that no missing values should exist, since dealing with missing values is outside the scope of this study. The datasets have a number of attributes ranging from 4 to 255 and number of instances ranging from 83 to 45211, with 8 datasets having less than 200 instances, 20 having 200 to 1000 instances, 10 having 1000 to 5000 instances and 4 having more than 5000 instances.

TABLE I. DATASETS INFORMATION

| # | Dataset | Attributes | Instances |
|---|---|---|---|
| 1 | Abalone | 8 | 4177 |
| 2 | Balance scale | 4 | 625 |
| 3 | Bank Full | 17 | 45211 |
| 4 | Banknote | 4 | 1372 |
| 5 | Blood Transfusion | 4 | 748 |
| 6 | Breast cancer (wdbc) | 32 | 569 |
| 7 | Breast cancer (winscon) | 9 | 699 |
| 8 | Breast_tissue | 9 | 106 |
| 9 | Cardiotocography | 21 | 2126 |
| 10 | Contraceptive | 9 | 1473 |
| 11 | Dermatology | 34 | 366 |
| 12 | Ecoli | 7 | 336 |
| 13 | Fertility Diagnosis | 9 | 100 |
| 14 | Firm Teacher | 19 | 10800 |
| 15 | Glass | 9 | 214 |
| 16 | GPS | 6 | 163 |
| 17 | Ionosphere | 34 | 351 |
| 18 | Iris | 4 | 150 |
| 19 | Messidor features | 18 | 1151 |
| 20 | Occupancy Detection | 7 | 20560 |
| 21 | Optdigit | 64 | 1797 |
| 22 | Parkinson disease | 22 | 195 |
| 23 | Parkinson speech | 26 | 1239 |
| 24 | Pendigits | 16 | 10992 |
| 25 | Pima indian | 8 | 768 |
| 26 | Seeds | 7 | 210 |
| 27 | Semeion | 255 | 477 |
| 28 | Sonar | 60 | 208 |
| 29 | Spect heart | 22 | 267 |
| 30 | Spectf heart | 44 | 267 |
| 31 | Statlog aust credit | 14 | 690 |
| 32 | Statlog german credit | 24 | 1000 |
| 33 | Statlog heart | 13 | 270 |
| 34 | Statlog img seg | 19 | 810 |
| 35 | Statlog sat image | 36 | 2000 |
| 36 | Thyroid disease new | 5 | 215 |
| 37 | Urban | 147 | 168 |
| 38 | Vertebral column 3C | 6 | 310 |
| 39 | Waveform | 21 | 5000 |
| 40 | Wine | 13 | 178 |
| 41 | Yeast | 8 | 1484 |
| 42 | Zoo | 16 | 83 |

For all experiments, the classification accuracy results per fold were obtained and a paired 10-fold cross validated T-test [30] was performed, in order to examine the statistical importance of the differences, with 95% confidence interval of 95%. In Table II, all datasets with statistically important difference are marked with a bullet in the right. The results presented in Table II indicate that the proposed approach is superior in terms of classification accuracy: the proposed algorithm outperforms the traditional K-means based CvC to all 42 datasets with the difference ranging from 1.14% to 43.22%. More detailed, the difference in 11 datasets is up to 5%, in 13 it ranges from 5% to 15%, while in another 10 datasets it’s between 15% and 25% and in the remaining 8 datasets it is over 25%. Accuracy differences for all datasets are illustrated in Figure 1. Regarding the average sensitivity results, once again the proposed algorithm outperforms simple K-Means to all datasets. The difference ranges from 1.96% to 32.16%. In 7 datasets it was below 5%, in 17 it was from 5% to 15%, in 12 datasets it was between 15% and 25%, while in 6 it surpassed 25%.

TABLE II. CLASSIFICATION-VIA-CLUSTERING RESULTS

| Dataset | | Classification Accuracy | | Average Sensitivity | | Average Precision | |
|---|---|---|---|---|---|---|---|
| | | **K-Means** | **C-K-Means** | **K-Means** | **C-K-Means** | **K-Means** | **C-K-Means** |
| 1 | Abalone | 50.73% | 52.21% | 50.90% | 53.54% | 52.57% | 50.84% |
| 2 | Balance scale ● | 55.52% | 68.64% | 51.43% | 73.93% | 55.03% | 71.66% |
| 3 | Banknote ● | 57.43% | 85.57% | 57.70% | 84.92% | 57.60% | 85.85% |
| 4 | Blood Transfusion ● | 59.09% | 66.44% | 49.20% | 67.16% | 49.35% | 62.75% |
| 5 | Breast cancer (Winsconsin) | 95.71% | 96.85% | 94.85% | 96.81% | 95.60% | 96.28% |
| 6 | Breast cancer (wdbc) | 92.62% | 93.85% | 91.05% | 93.28% | 93.15% | 93.53% |
| 7 | Breast_tissue ● | 41.51% | 60.38% | 45.02% | 58.62% | 50.40% | 61.40% |
| 8 | Cardiotocography ● | 34.20% | 54.99% | 38.32% | 64.02% | 34.30% | 53.60% |
| 9 | Contraceptive ● | 37.54% | 47.39% | 37.77% | 49.47% | 37.60% | 48.41% |
| 10 | Dermatology ● | 71.86% | 97.81% | 75.67% | 97.55% | 84.45% | 97.55% |
| 11 | Ecoli ● | 52.98% | 79.76% | 49.99% | 61.79% | 52.19% | 57.77% |
| 12 | Fertility Diagnosis ● | 44.00% | 68.00% | 50.40% | 60.23% | 50.20% | 54.96% |
| 13 | Firm Teacher ● | 55.56% | 98.78% | 65.80% | 97.96% | 58.80% | 97.62% |
| 14 | Glass | 40.19% | 45.33% | 40.27% | 51.96% | 29.32% | 52.39% |
| 15 | GPS ● | 66.87% | 82.82% | 65.15% | 81.66% | 71.05% | 87.05% |
| 16 | Ionosphere ● | 70.94% | 75.21% | 71.40% | 76.48% | 69.85% | 74.46% |
| 17 | Iris | 84.00% | 93.33% | 87.30% | 93.33% | 88.70% | 93.38% |
| 18 | Messidor Features | 52.74% | 57.41% | 51.85% | 57.42% | 52.00% | 57.39% |
| 19 | Optdigit ● | 74.80% | 90.76% | 75.81% | 90.76% | 76.24% | 91.13% |
| 20 | Parkinson Disease ● | 58.46% | 76.41% | 70.50% | 77.34% | 66.25% | 71.54% |
| 21 | Parkinson Speech | 57.78% | 60.02% | 57.35% | 60.04% | 57.25% | 59.85% |
| 22 | Pima Indian ● | 64.84% | 70.96% | 60.55% | 72.33% | 60.90% | 70.33% |
| 23 | Seeds | 88.57% | 91.43% | 88.57% | 91.43% | 88.90% | 91.38% |
| 24 | Semeion ● | 58.07% | 82.18% | 58.14% | 81.95% | 59.35% | 82.61% |
| 25 | Sonar ● | 54.33% | 73.56% | 54.35% | 73.08% | 54.35% | 73.68% |
| 26 | Spect Heart | 62.55% | 67.42% | 60.65% | 67.42% | 61.05% | 67.81% |
| 27 | Spectf Heart | 64.04% | 66.29% | 50.00% | 78.77% | 37.85% | 68.97% |
| 28 | Statlog Aust Credit ● | 76.81% | 85.65% | 78.55% | 86.30% | 76.90% | 85.91% |
| 29 | Statlog German Credit ● | 59.40% | 70.40% | 52.50% | 69.71% | 52.45% | 67.19% |
| 30 | Statlog Heart | 77.04% | 80.37% | 77.15% | 80.83% | 76.85% | 80.46% |
| 31 | Statlog Img Seg ● | 57.90% | 85.06% | 62.81% | 85.79% | 63.16% | 85.72% |
| 32 | Statlog Sat Image ● | 67.30% | 78.15% | 66.55% | 77.59% | 68.18% | 77.36% |
| 33 | Thyroid Disease New ● | 85.12% | 94.42% | 72.80% | 87.62% | 86.47% | 97.53% |
| 34 | Urban ● | 60.12% | 73.21% | 63.36% | 74.44% | 64.74% | 74.47% |
| 35 | Vertebral Column 3C ● | 44.19% | 74.52% | 41.52% | 73.22% | 43.17% | 72.23% |
| 36 | Waveform ● | 49.94% | 81.20% | 49.97% | 81.11% | 50.07% | 84.63% |
| 37 | Wine | 94.38% | 96.63% | 95.30% | 97.18% | 94.20% | 96.43% |
| 38 | Yeast ● | 36.86% | 50.27% | 37.93% | 57.24% | 34.00% | 48.65% |
| 39 | Zoo ● | 69.88% | 92.77% | 65.06% | 84.85% | 60.17% | 87.94% |
| 40 | Bank Full ● | 56.78% | 71.93% | 52.10% | 71.64% | 50.90% | 59.80% |
| 41 | Occupancy Detection ● | 62.16% | 87.99% | 68.60% | 89.36% | 63.25% | 82.48% |
| 42 | Pendigits ● | 70.87% | 82.94% | 71.51% | 82.93% | 72.54% | 83.20% |
| | ***Average*** | ***62.28%*** | ***76.41%*** | ***61.67%*** | ***76.44%*** | ***61.60%*** | ***75.45%*** |

Despite the smaller maximum difference, more datasets were observed where C-K-Means was found to outperform simple K-Means with a bigger difference. This indicates that C-K-Means classification managed to classify correctly more instances from each class (in average) than the traditional approach. All sensitivity results are also given in Figure 2. The last metric, average precision, is illustrated in Figure 3 and in detail in Table II. Again, the proposed approach outperforms traditional CvC. The difference ranges from 0.38% to 38.82%, while one dataset was found where the traditional approach presented slightly better precision (+1.73%). On the other hand, 10 datasets were found where C-K-Means classification was better up to 5%, 18 where it was better from 5% to 15%, 8 where the difference was from 15% to 25% and 6 where it was above 25%. Here is shown that the proposed approach is better than the traditional one in terms of if what is classified in a class actually belongs to that class. Once again, more datasets with higher differences to the benefit for the proposed algorithm are observed according to this metric. The paired 10-fold cross validated T-test revealed that the numerical difference in classification accuracy was statistically significant to 30 out of 42 datasets. Average sensitivity and precision results, alongside with the T-test validation in classification accuracy confirm in a versatile way that C-K-Means classification is superior to K-means based CvC.

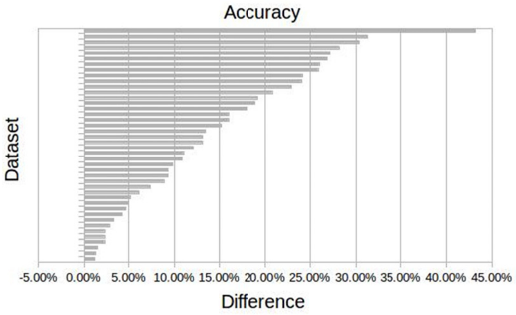

Fig. 1. Accuracy differences for all datasets

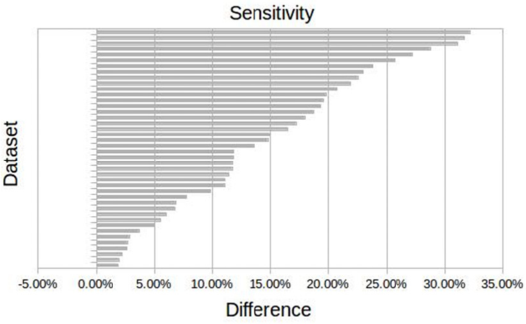

Fig. 2. Average sensitivity differences for all datasets

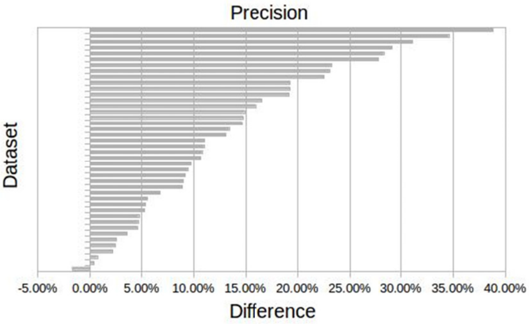

Fig. 3. Average precision differences for all datasets

## V. CONCLUSIONS

In this study, a novel CvC algorithm, based on K-Means clustering is proposed. The algorithms include two major modifications compared to the K-means, being (i) the use of a hypercube of constraints for each centroid extracted from the information of the training data, and (ii) the use weights for each attribute and each class along with the weighted Euclidean distance as a similarity criterion for the clustering procedure. An initial effort on this direction can be found in [31]. Since classification is a supervised learning method, the performance can be assessed by the classification rate, i.e. the number of successes and failures of the model. Reducing the number of misclassifications is very important regardless of which clustering algorithm or which variation of K-Means is used in CvC, an efficient use of the background knowledge is required to succeed. The introduction of constrains in the centroids initialization and update based on background knowledge, which is the main idea of the proposed algorithm, contributes to this direction. In the C-K-Means, both training (mainly associated with classification) and clustering are used, since by training the class hypercubes and attribute weights are obtained, and clustering is used for moving the centroids inside the hypercubes. Thus, the principal idea is to use the background knowledge to limit the solutions inside a predefined space (i.e. the hypercubes) and then follow an unsupervised technique to fine tune this solutions (by allowing the centroids to be updated but forcing them to remain inside the hypercube). Experimental results were obtained using 42 datasets from the UCI machine learning repository, with various number of attributes and instances. The results clearly demonstrate that the proposed approach dominates CvC with simple K-Means. Differences up to 43.22% in accuracy, 32.16% in average sensitivity and 38.82% in average precision are presented, while statistical analysis in terms of the paired 10-fold cross validated T-test revealed statistically significant differences in 30 out of 42 datasets. The proposed algorithm is generic, and in this realization K-means clustering and hard hypercube bound, have been used. Alternative approaches, such as fuzzy c-means or soft/ellipsoid bounds, can be integrated. Moreover, other types of weights or other metric of distance can be used as a similarity criterion. All the above will result to alternative realizations of the proposed algorithm; potential variations of the C-K-Means classification algorithm will be addressed in future communications.

## REFERENCES

[1] P. N. Tan, Introduction to data mining, Pearson Education India, 2006

[2] R. J. Schalkoff, Artificial neural networks. Vol. 1. McGraw-Hill, New York, 1997

[3] P. Langley, W. Iba, K. Thompson, “An analysis of Bayesian classifiers”, Tenth National Conference On Artificial Intelligence (AAAI-92), USA, July 12–16, 1992

[4] A. Chidanand, F. Damerau, S. M. Weiss, “Automated learning of decision rules for text categorization”, ACM Transactions on Information Systems (TOIS), Vol. 12, No. 3, pp. 233-251, 1994

[5] S. K. M. Wong, W. Ziarko, On optimal decision rules in decision tables, University of Regina, Computer Science Department, 1985

[6] J. R. Quinlan, “Induction of decision trees”, Machine Learning, Vol. 1, No. 1, pp. 81-106, 1986

[7] J. R. Quinlan, C4.5: Programming for machine learning, Morgan Kauffmann, 1993

[8] L. Rokach, “Ensemble-based classifiers”, Artificial Intelligence Review, Vol. 33, No. 1-2, pp. 1-39, 2010

[9] M. I. Lopez, J. M. Luna, C. Romero, S. Ventura, “Classification-via-clustering for predicting final marks based on student participation in forums”, International Educational Data Mining Society, 2012

[10] P. V. Gorsevski, P. E. Gessler, P. Jankowski, “Integrating a fuzzy k-means classification and a Bayesian approach for spatial prediction of landslide hazard”, Journal of Geographical Systems, Vol. 5, No. 3, pp. 223-251, 2003

[11] J. Erman, M. Arlitt, A. Mahanti, “Traffic classification using clustering algorithms”, SIGCOMM Workshop on Mining Network Data, Pisa, Italy, pp. 281-286, September 11-16, 2006

[12] M. Panda, M. R. Patra, “A novel Classification-via-clustering method for anomaly based network intrusion detection system”, International Journal of Recent Trends in Engineering, Vol. 2, No. 1, pp. 1-6, 2009

[13] P. A. Burrough, P. F. M. van Gaans, R. A. MacMillan, “High-resolution landform classification using fuzzy k-means”, Fuzzy Sets and Systems, Vol. 113, No. 1, pp. 37-52, 2000

[14] E. Reuben, B. Pfahringer, G. Holmes “Clustering for classification”, 7th International Conference on Information Technology in Asia, Malaysia, July 12-13, 2011

[15] S. Lloyd, “Least squares quantization in pcm”, IEEE Transactions on Information Theory, Vol. 28, No. 2, pp. 129-137, 1982

[16] J. MacQueen, “Some methods for classification and analysis of multivariate observations”, Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, Vol. 1, No. 14, 1967

[17] K. Wagstaff, C. Cardie, S. Rogers, S. Schrodl, “Constrained k-means clustering with background knowledge”, Proceedings of the Eighteenth International Conference on Machine Learning, pp. 577–584, Morgan Kaufmann, 2001

[18] I. Davidson, S. S. Ravi, “Clustering with constraints: Feasibility issues and the k-means algorithm”, Proceedings of the 2005 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2005

[19] P. S. Bradley, K. P. Bennett, A. Demiriz, “Constrained k-means clustering”, Microsoft Research, Redmond, pp. 1-8, 2000

[20] A. Likas, N. Vlassis, J. J. Verbeek, “The global k-means clustering algorithm”, Pattern Recognition, Vol. 36, No. 2, pp. 451-461, 2003

[21] J. C. Bezdek, R. Ehrlich, W. Full, “FCM: The fuzzy c-means clustering algorithm”, Computers & Geosciences, Vol. 10, No. 2-3, pp. 191-203, 1984

[22] D. Dua, K. E. Taniskidou, UCI Machine Learning Repository, Irvine, University of California, School of Information and Computer Science, 2017

[23] R. Payam, L. Tang, H. Liu, “Cross-validation” in: Encyclopedia of Database Systems, pp. 532-538, Springer, 2009

[24] G. Tsoumakas, I. Katakis, “Multi-label classification: An overview”, International Journal of Data Warehousing and Mining, Vol. 3, No. 3, 2006

[25] M. Inaba, K. Naoki, I. Hiroshi, “Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering”, Proceedings of the 10th ACM Annual Symposium on Computational Geometry, pp. 332-339, ACM, 1994

[26] P. S. Bradley, U. M. Fayyad, “Refining Initial Points for K-Means Clustering”, Proceedings of the 15th International Conference on Machine Learning (ICML98), pp. 91- 99, Morgan Kaufmann, 1998

[27] J. M. Pena, J. A. Lozano, P. Larranaga, “An empirical comparison of four initialization methods for the k-means algorithm”, Pattern Recognition Letters, Vol. 20, No. 10, pp. 1027-1040, 1999

[28] S. Kehar, D. Malik, N. Sharma, “Evolving limitations in K-means algorithm in data mining and their removal”, International Journal of Computational Engineering & Management, Vol. 12, pp. 105-109, 2011

[29] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, “The WEKA data mining software: an update”, ACM SIGKDD Explorations Newsletter, Vol. 11, No. 1, pp. 10-18, 2009

[30] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms", Neural Computation, Vol. 10, No. 7, pp. 1895-1923, 1998

[31] D. C. Tsouros, P. N. Smyrlis, M. G. Tsipouras, D. G. Tsalikakis, N. Giannakeas, A. T. Tzallas, P. Manousou, "Automated collagen proportional area extraction in liver biopsy images using a novel classification-via-clustering algorithm", IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), Greece, June 22-24, 2017