JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **11**, Number **3**, July **2015**

doi:10.3934/jimo.2015.11.921

pp. **921–932**

CLUSTERING BASED POLYHEDRAL CONIC FUNCTIONS ALGORITHM IN CLASSIFICATION

Gurkan Ozturk

Department of Industrial Engineering, Faculty of Engineering, Anadolu University
Eskisehir, 26555, Turkiye

Mehmet Tahir Ciftci

Vitra
Eczacibasi Yapi Gerecleri
11300 Bilecik, Turkiye

(Communicated by Renata Sotirov)

Abstract. In this study, a new algorithm based on polyhedral conic functions (PCFs) is developed to solve multi-class supervised data classification problems. The k PCFs are constructed for each class in order to separate it from the rest of the data set. The k-means algorithm is applied to find vertices of PCFs and then a linear programming model is solved to calculate the parameters of each PCF. The separating functions for each class are obtained as a pointwise minimum of the PCFs. A class label is assigned to the test point according to its minimum value over all separating functions. In order to demonstrate the performance of the proposed algorithm, it is applied to solve classification problems in publicly available data sets. The comparative results with some mainstream classifiers are presented.

1. **Introduction.** Supervised data classification problems are among most important in data mining and machine learning. They have lots of applications including banking, drug design, engineering, business, medicine and voice recognition. There are different approaches for solving this problem and mathematical programming approaches are among most powerful. For example, the well known Support Vector Machines algorithms involve solving of a quadratic programming problem (see, for example, [8, 16]).

Mathematical programming approaches allow to consider different nonlinear separating surfaces. Such approaches include h-polyhedral separability [1], max-min separability [4, 5] and polyhedral conic separability [7, 9] which provide piecewise linear separating surfaces between classes. In all cases some kind of nonsmooth error function is formulated and minimized using optimization and in particular, nonsmooth derivative free optimization techniques [6].

The concept of polyhedral conic separability based on polyhedral conic functions (PCFs) was introduced in [9](see, also [14]). An algorithm for calculation of polyhedral conic functions separating two sets was developed in [9]. This algorithm randomly chooses a data point from one of these sets as a first vertex and computes

2010 *Mathematics Subject Classification.* Primary: 68Q32, 46N10; Secondary: 97R50.

Key words and phrases. Classification , polyhedral conic cunctions, K-means, linear programming, computational learning theory.

the first PCF. Then all data points from this set separated by the obtained PCF are removed from the set and next vertex is randomly selected from the rest of the set. This process continues until all points from the selected set are separated. A classifier is constructed as a pointwise minimum of all obtained PCFs. Despite some promising results such an approach may suffer over-fitting. Another algorithm was introduced in [14] based on the bi-objective integer programming approach. Objectives in this approach are to minimize the number of PCFs separating sets and to maximize the number of correctly classified points. Although this algorithm suffers over-fitting problem in some data sets, however it reduces this problem in comparison with the first algorithm. Furthermore this algorithm is time consuming in large data sets.

In this paper, a new classifier is designed based on the combination of the polyhedral conic separation approach and k-means clustering technique. We apply k-means algorithm to find vertices of PCFs and then find PCFs for each cluster by solving a linear programming problem. This classifier is different from that given in [9, 14] where the final classifier is obtained by sequentially eliminating the correctly classified points whereas in the proposed algorithm the classifier is constructed in one step using cluster centers found by the k-means algorithm. The use of clustering algorithms allows to decrease significantly the number of vertices and consequently the number of PCFs which helps to avoid over-fitting problem. Moreover the use of linear programming techniques makes the algorithm applicable to large data sets.

The proposed algorithm is applied to solve classification problems in some publicly available data sets including large data sets. We demonstrate the performance of the algorithm using one illustrative example. Comparison of the proposed classifier with some mainstream classifiers is also presented using numerical results.

The rest of the paper is organized as follows. Section 2 provides a brief description of polyhedral conic separation and k-means algorithm. The proposed algorithm is given in Section 3 and computational results of this algorithm are reported in Section 4. Some concluding remarks are given in Section 5.

2. Preliminerales.

In this section we briefly describe the concept of polyhedral conic separability as well as the k-means algorithm for clustering.

2.1. Polyhedral conic separation.

PCFs have recently been proposed to construct a separation function for the finite point sets $A, B \subset R^n$ [9]. Definition 2.1 and Lemma 2.2 quoted below are given in [9]. The notion of conic separation is also discussed in [2, 12, 11, 13].

Definition 2.1. A function $g : \mathbb{R}^n \to \mathbb{R}$ is called *polyhedral conic* if its graph is a cone and all its level sets

$$S(\alpha) = \{x \in \mathbb{R}^n : g(x) \leq \alpha\},$$

for $\alpha \in \mathbb{R}$, are polyhedrons.

Figures 1 and 2 illustrate the graph and level set of the following polyhedral conic function,

$$g(x_1, x_2) = 0.11(x_1 - 2) + 0.11(x_2 - 2) + 0.33(|x_1 - 2| + |x_2 - 2|) - 1$$

respectively.

Given $w, c \in \mathbb{R}^n, \xi, \gamma \in \mathbb{R}$ a PCF $g_{(w,\xi,\gamma,c)}$: $\mathbb{R}^n \to \mathbb{R}$ is defined as follows:

$$g_{(w,\xi,\gamma,c)}(x) = \langle w, x - c\rangle + \xi \|x - c\|_1 - \gamma, \tag{1}$$

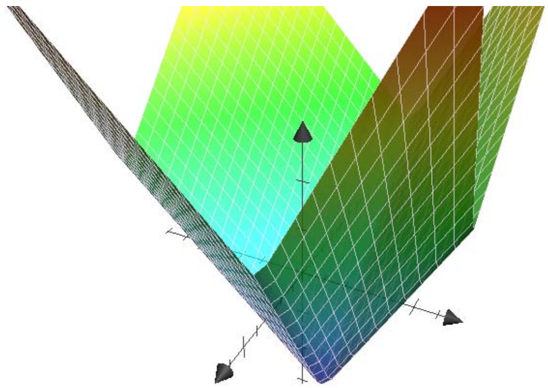

Figure 1. The graph of the polyhedral conic function g

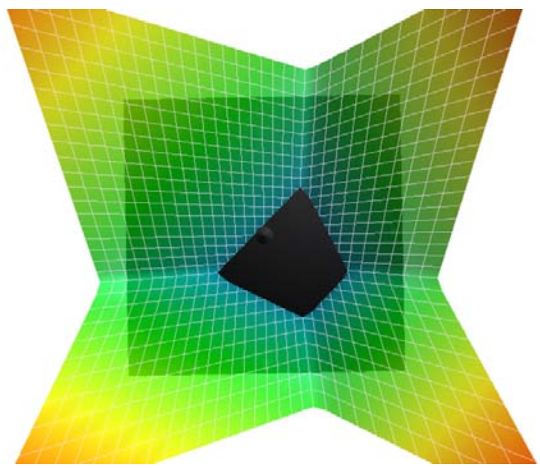

Figure 2. The level set of the polyhedral conic function g

where $\|x\|_1 = |x_1| + \cdots + |x_n|$ is an l_1-norm of the vector $x \in \mathbb{R}^n$ and $\langle \cdot, \cdot \rangle$ is an inner product in $\mathbb{R}^n$.

Lemma 2.2. *A graph of the function $g_{(w,\xi,\gamma,c)}$ defined in (1) is a polyhedral cone with a vertex at $(c, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$.*

The sets A and B are polyhedral conic separable if there exist a finite number of PCFs $g_l = g_{(w^l,\xi^l,\gamma^l,c^l)}$, $l = 1, \ldots, L$ such that

$$\min_{l=1,\ldots,L} g_l(a) \leq 0 \ \forall\, a \in A$$

and

$$\min_{l=1,\ldots,L} g_l(b) > 0 \ \forall\, b \in B.$$

The way a PCF can separate two sets A and B in $\mathbb{R}^2$ is shown in Figure 3. In this figure the set A contains red points and the set B contains blue points. The graph of the PCF separating these two sets is in yellow. Even though A and B are linearly inseparable, the constructed polyhedral conic function can completely

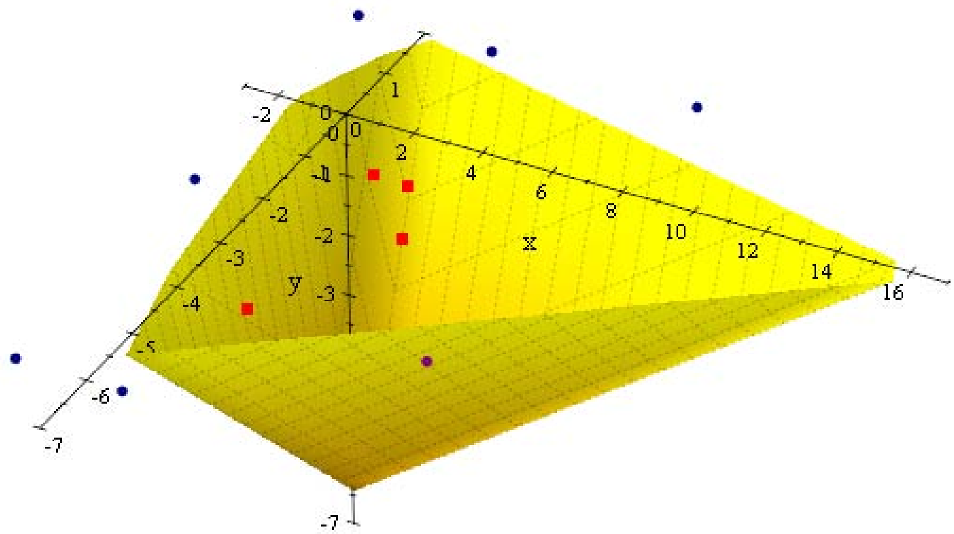

FIGURE 3. Separation using a polyhedral conic function $g(x,y) = -1.9x + 1.1y + 2.3\,(|x| + |y|) - 6.6$

separate these sets. The level set of the separating function is the intersection of four half-spaces: $0.4x + 4.4y - 6.6 \leq 0$, $0.4x - 1.2y - 6.6 \leq 0$, $-4.2x - 1.2y - 6.6 \leq 0$ and $-4.2x + 4.4y - 6.6 \leq 0$. An algorithm generating a polyhedral conic separating function, therefore called a PCF algorithm, is developed in [9].

2.2. **K-means.** K-means algorithm is one of the well known algorithms for solving clustering problems. It is easy to implement, fast and applicable to very large data sets.

Algorithm 1. K-Means Algorithm

Step 1. Choose a seed solution consisting of k centers (not necessarily belonging to A).

Step 2. Allocate data points $a \in A$ to their closest centers and obtain k-partition of A.

Step 3. Recompute centers for this new partition and go to Step 2 until no more data points change their clusters.

3. **The K-means PCFs algorithm.** In this section we design a new classifier for solving supervised classification problems in finite point data sets with finite number of classes. This algorithm applies k-means algorithm to find vertices of PCFs.

Assume we are given finite point set $A \subset R^n$ with p classes. More specifically the set

$$A = \{a^i \in R^n : i \in I\}, \quad I = \{1, 2, \ldots, m\}$$

and its classes $A_j, j = 1, \ldots, p$ are given. For each class A_j we construct the following set

$$B_j = \bigcup_{l=1, l\neq j}^{p} A_l$$

which is a union of all classes except the class A_j.

Algorithm 2. **k-means PCF**

Step 0. Set $j := 0$ and select the number of clusters, k.
Step 1. Set $j := j + 1$ and select the sets A_j and B_j.
Step 2. Apply the k-means algorithm to the set A_j to find k clusters A_{jr} and their centroids $c_{jr} \in R^n$, $r = 1, \ldots, k$.
Step 3. Find the k-PCFs, g_{jr}, $r = 1, \ldots, k$, with the parameters $(w^{jr}, \xi_{jr}, \gamma_{jr})$ for class j by solving the following linear programming problem (P_{jr}) for each cluster $A_{jr}, r = 1, \ldots, k$.

$$(P_{jr}) \quad \begin{array}{ll} & \min \frac{1}{|A_{jr}|} \sum_{i \in I_{jr}} y_i + \frac{1}{|B_j|} \sum_{l \in I_B} z_l \\ \text{subject to} & \\ & w^{jr}(a_i - c_{jr}) + \xi_{jr} \|a_i - c_{jr}\|_1 - \gamma_{jr} \le y_i, \qquad \forall\, i \in I_{jr} \\ & -w^{jr}(a_l - c_{jr}) - \xi_{jr} \|a_l - c_{jr}\|_1 - \gamma_{jr} \le z_l, \qquad \forall\, l \in I_B \end{array}$$

where $I_{jr} = \{i : a_i \in A_{jr}\}$ and $I_B = \{i : a_i \in B_j\}$.
Step 4. Construct the separating function for the class j as follows:

$$g_j(x) = \min_{r=1,\ldots,k} g_{jr}(x).$$

Step 5. If $j < p$ go to *Step 1*, otherwise the algorithm terminates.

3.1. **Illustrative example.** In order to demonstrate how the proposed algorithm works, we present an illustrative example with the data set given in two dimensional space, R^2. This data set is presented in Figure 4 and contains three classes shown as green, yellow and purple. As one can see from the figure convex hulls of yellow and purple classes have a big overlap. Convex hull of the green class contains most of data set and moreover this class has two different subsets whose convex hulls are disconnected. All these circumstances make solution of supervised data classification problem in this data set challenging for many existing classifiers.

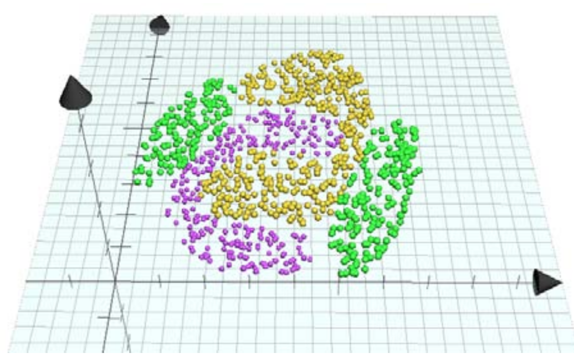

FIGURE 4. Data points in R^2 for the illustrative example

According to notation used for description of Algorithm 2 in this illustrative example

$$A = \{a^i \in R^2 : i \in I\}, I = \{1, 2, \ldots, 998\}$$

$p = 3$, $A = \bigcup_{j=1}^{3} A_j$ and A_1 is green, A_2 is purple and A_3 is yellow class.

In Step 0 we select the number of clusters $k = 5$. In order to construct the classifier we have to find the separating function for each class. We demonstrate finding such separating function only for one class, namely, for class A_1. According to Step 1, $j = 1$ and $B_1 = A_2 \bigcup A_3$. Implementation of this step is illustrated in Figure 5. In Step 2 the k-means algorithm is applied to find five clusters A_{1r} in A_1 and their centroids $c_{1r}, r = 1, \dots, 5$. In Step 3 for each cluster A_{1r} one PCF g_{1r} is computed by solving the linear programming problem (P_{1r}) using points from A_{1r} and B_1 and centroid $c_{1r}, r = 1, \dots, 5$. Results of Steps 2 and 3 are illustrated in Figure 6 where cluster centers are shown by "+". Graphs of functions g_{1r} for each cluster A_{jr} are given in Figure 7 and their graphs for the whole class A_1 are depicted in Figure 8. The separating function g_1 for the class A_1 is computed as pointwise minimum of PCFs g_{1r} in Step 4. This function is illustrated in Figure 9.

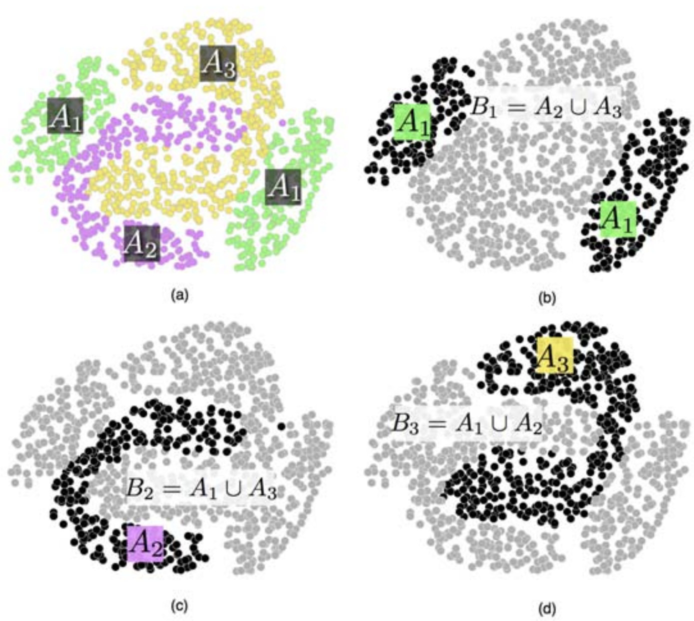

Figure 5. Implementation of Step 1

3.2. **Analysis of the proposed algorithm.** In this section three types of analysis about performance of the proposed algorithm are presented. Namely, we analyze: (i) the dependence of performance of the algorithm on the choice of initial centers in the k-means algorithm; (ii) the dependence of generalization ability of the algorithm on the number of clusters; (iii) comparison of the algorithm with other PCF based classifiers. For these analysis, we used four small sized data sets from the UCI-ML Repository namely, Iris Plant, Vehicle, Wine and Vowel data sets. Performance of the algorithm for medium and large scale data sets will be presented in Section "Computational Results".

Effects of initial center points are investigated on Iris Plant data set. Ten experiments are performed by using different randomly selected initial center points and obtained training and test set accuracy results are presented in Table 1. These results confirm that the performance of the algorithm is not sensitive to the choice of initial centers in the k-means algorithm.

In order to show how the generalization ability of the proposed algorithm depends on the number of clusters, this algorithm is applied to the Vehicle data set with

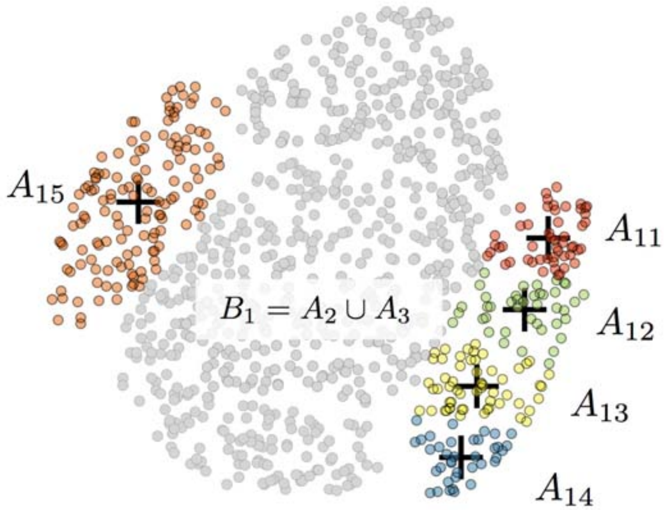

FIGURE 6. Implementation of Steps 2-3

TABLE 1. Effects of randomly selected initial center points on the performance of the algorithm

											average
Test	97.33	95.33	96.67	96.00	96.00	96.00	96.00	96.67	95.33	96.00	**96.13**
Training	98.96	97.93	97.78	98.89	98.22	97.63	97.70	98.74	98.15	97.41	**98.14**

TABLE 2. Effects of number of clusters over

k	2	3	4	5	6	7	8	9	10	20
training	86.01	87.84	89.81	90.50	91.75	92.30	93.22	93.67	94.29	98.59
test	81.80	81.31	83.44	84.05	80.97	80.38	79.78	78.96	78.35	72.92
generalization%	4.90	7.43	7.09	7.13	11.76	12.92	14.42	15.70	16.90	26.04

different number of clusters k. The obtained result are presented in Table 2. In this table the row *generalization%* is calculated by the formula $(trainsetaccuracy - testsetaccuracy)/trainsetaccuracy$. These results show that as the number of PCFs (or clusters) increases the training set accuracy approaches to 100%. However, at the same time the difference between training and test sets accuracies also increases. This means that as the number of PCFs (clusters) increases the generalization ability of the algorithm worsens and therefore this number cannot be large.

In order to determine the number of clusters for each data set, its training set is divided into a new training and test sets. Then the classification subproblem is solved using the number of clusters less than a given number. The number of clusters with the best test set accuracy is chosen to solve the original classification problem. Note that the similar methodology is also used to determine parameters in SVM.The comparison of the proposed algorithm with other PCF based classifiers [14] are made on the small sized data sets and obtained results are presented in Table 3. From these results, one can see that generalization performance of the

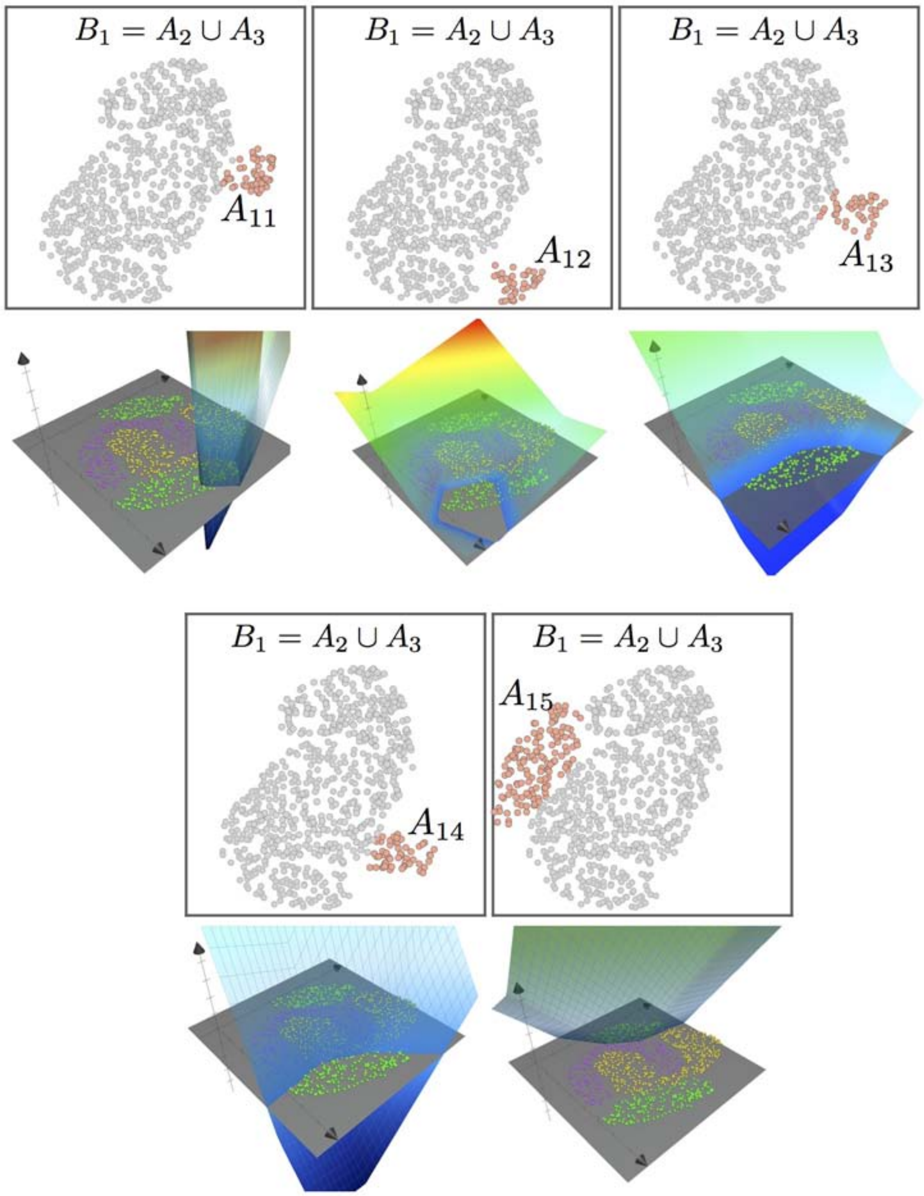

FIGURE 7. Graphs of PCFs for each cluster A_{1r}

proposed algorithm is improved comparing to PCF algorithm [14] and Bi-objective Integer Programming Approach [14] for all data sets with the exception of the Vowel data set. For this data set the bi-objective integer programming classifier provides better test set accuracy and generalization performance.

4. **Computational results.** The efficiency of the proposed algorithm is tested using data sets publicly available in UCI-ML Repository [3]. We used four data sets, namely, Abalone, Shuttle Control, PageBlocks and Letter Recognition data

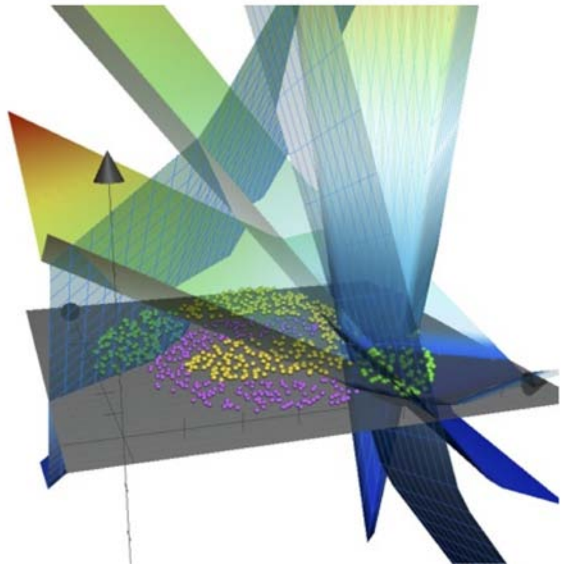

FIGURE 8. Graphs of all PCFs for class A_1

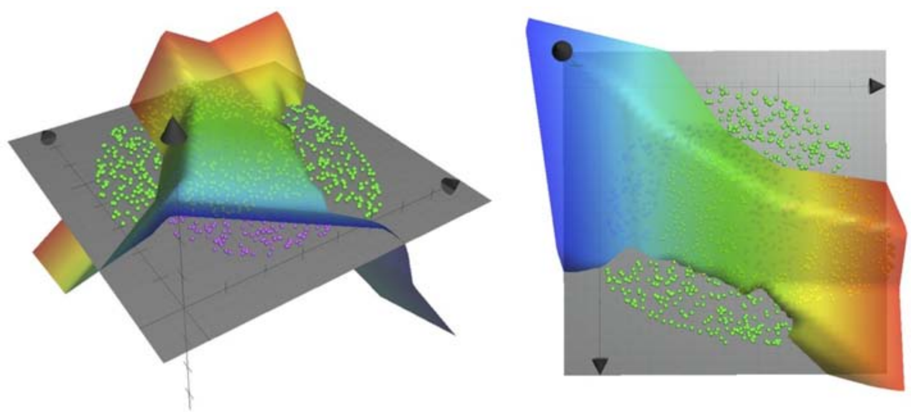

FIGURE 9. Separating function g_1 for A_1

TABLE 3. The comparison of different PCF based classifiers

data set	k-means PCF		PCFA [14]		Int. Prog. [14]	
iris	96.67	97.7	100	96.67	97.33	98.59
vehicle	90.5	84.05	100	78.28	94.85	79.79
wine	100	97.22	100	96.63	100	96.63
vowel	99.05	90	100	87.12	99.87	92.05

sets. Shuttle and Letter Recognition data sets are large whereas other two data sets can be considered as medium size data sets. Brief description of the data sets is given in Table 4 where the number of data points for training and test sets, number of classes and number of attributes for each data sets are included. More

Table 4. The brief description of data sets

	Shuttle	**Letters**	**Page-blocks**	**Abalone**
# data points	58000	20000	5473	4177
# points in training set	43500	15000	4000	3133
# points in test set	14500	5000	1473	1044
# class	7	26	5	3
# attributes	10	17	11	9

Table 5. Classification accuracy for the training and test sets obtained by k-means-PCF

	Data set	Abalone	Shuttle	Letter	Page-Block
K-means-PCF	Training	64.70	99.64	89.70	97.95
	Test	64.27	99.69	84.28	92.87

detailed description of these data sets can be found [3]. The data sets given in Table 4 have only real or integer attributes and there is no missing value in them.

k-means-PCF algorithm is implemented in General Algebraic Modeling System (GAMS). GAMS is a high-level modeling system for mathematical programming and optimization. It consists of a language compiler and a stable of integrated high-performance solvers [15]. The linear programming problem in step 3 of Algorithm 2 is solved using CPLEX solver from GAMS.

Classification accuracy for the training and test sets obtained by the k-means-PCF algorithm are presented in Table 5. One can see that the proposed algorithm demonstrate good generalization ability on two data sets: Abalone and Shuttle Control. These two data sets have good cluster structure which was detected by k-means algorithm. However in other two data sets k-means algorithm is not so successful in finding clusters which leads to some over-fitting problem. This means that the generalization ability of the proposed algorithm strongly depends on success of the clustering algorithm. However the generalization ability of this algorithm is significantly better than that of PCF algorithm introduced in [9].

The proposed algorithm is compared with the number of classifiers from data mining package WEKA. WEKA is a popular machine learning suite for data mining tasks written in Java and developed at the University of Waikato, New Zealand (see for details [10]). We select Naive Bayes (NB), Logistic, Multi Layer Perception (MLP), Support Vector Machine (SMO) , SMO(NPOL), SMO(PUK) classifiers for comparison. Results are presented in Table 6. One can see that the proposed classifier produce better test accuracy than any other classifier in three data sets: Abalone, Letters and Page Blocks. This classifier generates the second best result in Shuttle Control data set.

We also tested the proposed classifier using data set from Subsection 3.1 by applying ten fold cross validation. The test accuracy results produced by different classifiers are presented in Table 7. These results clearly demonstrate that the proposed classifier considerably outperforms others in solving classification problems in data sets with disconnected and overlapped classes.

5. **Conclusions.** In this paper we developed new classifier based on polyhedral conic functions. This algorithm is the combination of the k-means clustering algorithm and polyhedral conic separation. The k-means algorithm is applied to find

TABLE 6. Comparison of results obtained by k-means-PCF and other classifiers

	Abalone	Shuttle	Letters	Page-Block
NB	57.85	98.32	74.12	88.39
Logistic	**64.27**	96.83	77.40	91.72
MLP	63.51	**99.75**	83.20	92.80
SMO	60.73		82.40	87.03
SMO(NPOL)	60.25	96.81	82.34	89.48
SMO(PUK)	64.18	99.50		88.53
k-means-PCF	**64.27**	99.69	**84.28**	**92.87**

TABLE 7. Comparison of ten fold cross validation results on synthetic data set

	Test Accuracy
SMO	56.21
Naive Bayes	76.45
Logistic	53.20
k-means-PCF	98.98

vertices of the polyhedral conic functions. In order to find the separating function based on polyhedral conic functions, this approach reduce minimization of the non-convex nonsmooth error function to the solving of number of linear programming problems. The use of clustering algorithms allows to improve generalization ability of classifiers based on polyhedral conic separation by reducing the number of polyhedral conic functions. The efficiency of the proposed classifier is demonstrated using four real world and one synthetic data sets. The synthetic data set contains classes whose convex hulls have large overlaps and one class consists of two parts whose convex hulls are disconnected. This makes classification problem in such data sets challenging for many existing classifiers. Results for real world data sets show that the proposed classifier performs similar to well known mainstream classifiers. Test results on synthetic data set demonstrate that the proposed classifier significantly outperforms other well known classification algorithms in such data sets. The k-means-PCF classifier requires low memory and almost no testing time that is this classifier is real time classifier and can be applied to solve classification problems in many real world applications where these two requirements are crucial.

Acknowledgments. The authors would like to thank two anonymous referees for their criticism and comments which allowed to improve the quality of the paper. The authors also thank Mr. Emre Cimen for his help in coding the proposed algorithm. This study was supported by Anadolu University Scientific Research Projects Commission under the grant no:1103F035.

REFERENCES

[1] A. Astorino and M. Gaudioso, Polyhedral separability through successive LP, *Journal of Optimization Theory and Applications*, **112** (2002), 265–293.

[2] A. Astorino, M. Gaudioso and A. Seeger, Conic separation of finite sets. i: The homogeneous case, *Journal of Convex Analysis*, **21** (2014), 001–028.

[3] K. Bache and M. Lichman, UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

[4] A. M. Bagirov, Max-min separability, *Optimization Methods and Software*, **20** (2005), 277–296.

[5] A. M. Bagirov and J. Ugon, Supervised data classification via max-min separability, *Applied Optimization*, **99** (2005), 175–207.

[6] A. M. Bagirov, M. Ghosh and D. Webb, A derivative-free method for linearly constrained nonsmooth optimization, *Journal of Industrial and Management Optimization*, **2** (2006), 319–338.

[7] A. M. Bagirov, J. Ugon, D. Webb, G. Ozturk and R. Kasimbeyli, A novel piecewise linear classifier based on polyhedral conic and max-min separabilities, *TOP*, **21** (2013), 3–24. ISSN 1134-5764.

[8] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2** (1998), 121–167.

[9] R. N. Gasimov and G. Ozturk, Separation via polihedral conic functions, *Optimization Methods and Software*, **21** (2006), 527–540.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The weka data mining software: An update, *SIGKDD Explorations*, **11** (2009), 10–18. ISSN 1931-0145.

[11] R. Kasimbeyli, Radial epiderivatives and set-valued optimization, *Optimization*, **58** (2009), 521–534.

[12] R. Kasimbeyli, A nonlinear cone separation theorem and scalarization in nonconvex vector optimization, *SIAM J. on Optimization*, **20** (2009), 1591–1619. ISSN 1052-6234.

[13] R. Kasimbeyli and M. Mammadov, On weak subdifferentials, directional derivatives, and radial epiderivatives for nonconvex functions, *SIAM Journal on Optimization*, **20** (2009), 841–855.

[14] G. Ozturk, *A New Mathematical Programming Approach to Solve Classification Problems*, PhD thesis, Eskisehir Osmangazi University, Institute of Scince, **6** 2007. (in Turkish).

[15] R. Rosenthal, *GAMS: A User's Guide*, GAMS Development Corporation, Washington, DC, 2013. URL http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf.

[16] K. Schittkowski, Optimal parameter selection in support vector machines, *Journal of Industrial and Management Optimization*, **1** (2005), 465–476.

Received October 2013; 1st revision May 2014; 2nd revision June 2014.

E-mail address: gurkan.o@anadolu.edu.tr
E-mail address: tahir.ciftci@eczacibasi.com.tr