Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1172 – 1180
© TÜBİTAK
doi:10.3906/elk-1806-45

Polyhedral conic kernel-like functions for SVMs

Gürkan ÖZTÜRK[1,2,*], **Emre ÇİMEN**[1,2]
[1]Department of Industrial Engineering, Engineering Faculty, Eskişehir Technical University, Eskişehir, Turkey
[2]Computational Intelligence and Optimization Laboratory, Engineering Faculty, Eskişehir Technical University, Eskişehir, Turkey

Received: 08.06.2018 • **Accepted/Published Online:** 17.12.2018 • **Final Version:** 22.03.2019

Abstract: In this study, we propose a new approach that can be used as a kernel-like function for support vector machines (SVMs) in order to get nonlinear classification surfaces. We combined polyhedral conic functions (PCFs) with the SVM method. To get nonlinear classification surfaces, kernel functions are used with SVMs. However, the parameter selection of the kernel function affects the classification accuracy. Generally, in order to get successful classifiers which can predict unknown data accurately, best parameters are explored with the grid search method which is computationally expensive. We solved this problem with the proposed method. There is no need to optimize any parameter in the proposed method. We tested the proposed method on three publicly available datasets. Next, the classification accuracies of the proposed method were compared with the linear, radial basis function (RBF), Pearson universal kernel (PUK), and polynomial kernel SVMs. The results are competitive with those of the other methods.

Key words: Support vector machines, polyhedral conic functions, kernel functions, classification, mathematical programming

1. Introduction

Solving classification problems efficiently is very important for real-life applications in the areas of machine learning and data mining. Therefore, many researchers have been actively working on this problem. One of the most popular methods for solving this problem is support vector machines (SVMs). SVMs control generalization error by balancing the separation margin and the misclassification rate.

The first version of the SVMs constructs a linear hyperplane to separate classes. However, a linear classifier does not meet the needs of a user because the most of the real-life classification problems are not linearly separable. In order to cope with this problem, SVMs use kernel functions to get nonlinear separating surfaces. Selection of the kernel function and determination of its parameters play a crucial role in the accuracy.

Polyhedral conic functions (PCFs) have been introduced to solve classification problems [1]. These functions can be obtained by extending linear functions with l_1-normed term. There are various algorithms based on PCFs. These algorithms are applied to some real-life problems like arrhythmia classification and gesture recognition. The parameters of the PCFs are found by solving an optimization problem.

In [2], the advantages of PCFs and SVMs were combined and an algorithm for binary classification problem was proposed. Unlike [2], our study has been developed for both binary and multiclass classification problems. Moreover, our aim was different and we used PCFs as a kernel-like function to improve classification accuracy of

*Correspondence: gurkan.o@eskisehir.edu.tr

the SVMs. Firstly, a PCF that represents the training data was found by solving a linear programming problem for each class and then a feature was added to the training set by calculating the corresponding PCF value. This means that the dimension of the original data is increased by the number of classes.

The notation we use in this study is as follows. The vectors are represented by **boldface** lower-case letters, $\mathbf{a}$. Scalars are represented by *italic* lower-case letters, s. The $2d$ matrices which are the datasets in this paper are symbolized by capital **boldface** letters, $\mathbf{A}$. For the cardinality of a set, the $|.|$ is used, and the number of classes is represented by the symbol η.

In the following sections, SVMs and PCFs are described briefly. In Section 4, the proposed algorithm with an illustrative example is given. Section 5 gives some computational results, and Section 6 concludes the study.

2. Support vector machines

SVMs are famous supervised learning models in machine learning which were introduced in [3]. They are defined for binary classification problems. However, it can be generalized to multiclass classification problems. For linear separable case, a lot of separating hyperplane which separates two sets completely can be found with different methods. Each of these hyperplanes can be used as a decision boundary between classes. In the SVM method, the objective is to find the separating hyperplane which has the maximum distance to both classes. The distance between the separating hyperplane and closest data point to the hyperplane is called margin.

It is often unlikely to be encountered with linearly separable case for many real-life problems. In the linearly inseparable cases, SVMs use a kernel function in order to obtain nonlinear decision boundaries. This method is called kernel trick and was proposed in [4]. Gaussian kernel, polynomial kernel, and sigmoid kernel are the common examples of kernel functions. In Figure 1, how to obtain a nonlinear separation surface with the Gaussian kernel in SVM for a linearly nonseparable case is shown. Though kernel functions improve classification success, it is very important to select the right kernel function and its parameters. Generally, the best parameters are found by a method called grid search.

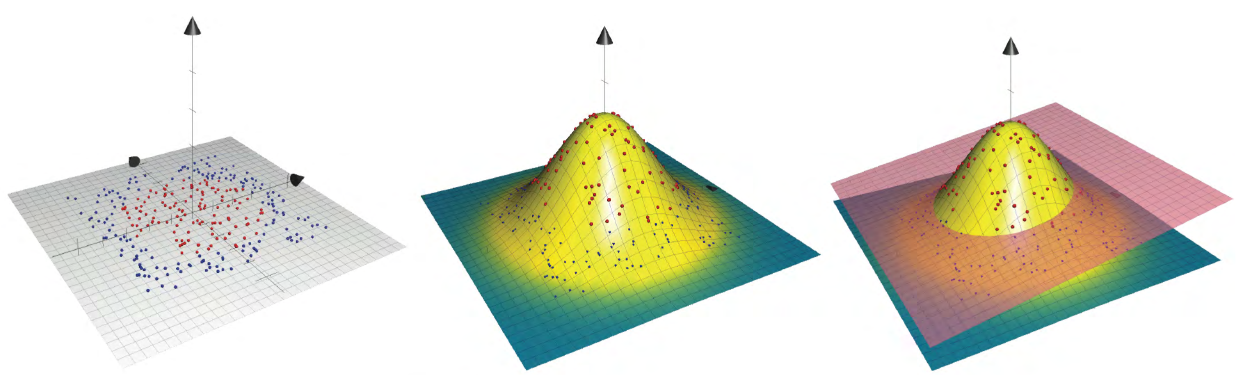

Figure 1. An example for RBF kernel.

Kernel functions allow researchers to apply SVMs on real-life problems such as text classification [5] and biology [6]. Multiple kernel learning algorithms have had great impact in recent years [7].

The following kernel functions are commonly used:

- Gaussian radial basis kernel (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$,

- Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + c)^h$,
- Pearson universal kernel (PUK): $K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\left[1+\left(\frac{2\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\sqrt{2^{(1/w)}-1}}{\sigma}\right)^2\right]^w}$.

The SVM model requires solving an optimization problem. It can be summarized as follows [3, 8, 9]:

$$\begin{array}{lrcl} \min\limits_{\mathbf{w},b,\boldsymbol{\xi}} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum\limits_{i\in I}\xi_i & & \\ \text{s.t.} & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) & \geq & \xi_i, \\ & \xi_i & \geq & 0, \end{array} \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ is the training data points, $y_i \in \{-1, 1\}$ is the associated class labels, and $i \in I = \{1, \ldots, m\}$. By solving the SVM model, a hyperplane is obtained with the normal vector $\mathbf{w}$ and b parameters. $\phi(.)$ is the kernel and maps the input vector into a higher dimensional space. $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. ξ_i corresponds misclassification error of i^{th} data point. Thus, as one can see from the constraints, $\xi_i = 0$ if i^{th} data point is classified correctly. Otherwise, ξ_i gets a positive value. The objective function has two parts; while the first part tries to maximize the margin, the second part tries to minimize the classification error. Therefore, there is a trade-off between these two objectives.

3. Polyhedral conic functions and separation

A polyhedral conic function (PCF) is obtained by adding l_1 norm term with a positive scalar coefficient, τ to the linear function. Eq. 2 shows that a PCF where $\mathbf{w}, \mathbf{x} \in \mathbb{R}^\mathbf{n}$, and $\gamma, \tau \in \mathbb{R}^+$, $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$.

$$g(\mathbf{x}) = \mathbf{w}'\mathbf{x} + \tau\|\mathbf{x}\|_1 - \gamma. \tag{2}$$

A vertex point of a PCF, which is defined as in Eq. 2, is composed of a center point (origin, $\mathbf{0}$) and positive scalar γ as $(\mathbf{0}, -\gamma)$. In order to construct a PCF with a specific center, $\mathbf{c} \in \mathbb{R}^n$, the function can easily be written with the form given in Eq. 3.

$$g(\mathbf{x}) = \mathbf{w}'(\mathbf{x} - \mathbf{c}) + \tau\|\mathbf{x} - \mathbf{c}\|_1 - \gamma. \tag{3}$$

PCFs were proposed to construct a separation function for the finite point sets $\mathbf{A}, \mathbf{B} \subset \mathbb{R}^n$ [1]. Definition 1 and Lemma 1 quoted below were given in [1].

$$\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}, \quad \mathbf{a}_i \in \mathbb{R}^n, \quad i \in I = \{1, \ldots, m\} \tag{4}$$

$$\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_p\}, \quad \mathbf{b}_j \in \mathbb{R}^n, \quad j \in J = \{1, \ldots, p\} \tag{5}$$

Definition 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*

$$S(\alpha) = \{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) \leq \alpha \}, \tag{6}$$

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

Lemma 1 *A graph of the function* $g_{(w,\xi,\gamma,c)}$ *defined in Eq. (2) is a polyhedral cone with a vertex at* $(c, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$.

In order to obtain PCF parameters $\mathbf{w}$, τ, and γ; it is required to solve a linear programming problem, P.

$$(P) \quad \begin{array}{l} min \frac{1}{|\mathbf{A}|} \sum_i \xi_i^A + \frac{1}{|\mathbf{B}|} \sum_j \xi_j^B \\ \text{s.t.} \\ \ \mathbf{w}(\mathbf{a}_i - \mathbf{c}) + \tau \|\mathbf{a}_i - \mathbf{c}\|_1 - \gamma \leq \xi_i^A, \forall\ i \in I \\ -\mathbf{w}(\mathbf{b}_j - \mathbf{c}) - \tau \|\mathbf{b}_j - \mathbf{c}\|_1 - \gamma \leq \xi_j^B, \forall\ j \in J \end{array}$$

Figures 2 and 3 illustrate the graph and level set of the polyhedral conic function, respectively.

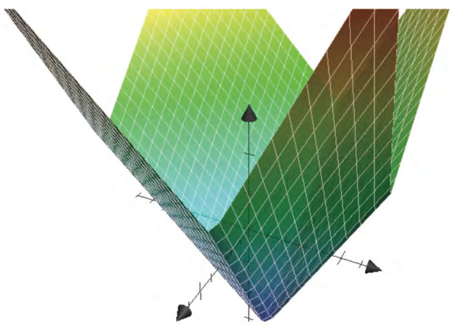

Figure 2. The graph of the polyhedral conic function g.

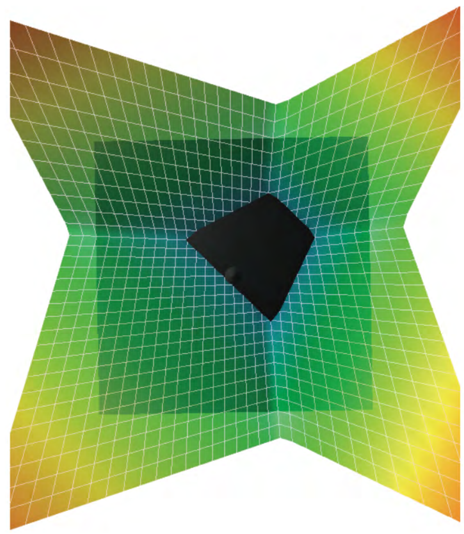

Figure 3. The level set of the polyhedral conic function g.

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

$$\min_{l=1,\ldots,L} g_l(\mathbf{a}) \leq 0 \quad \forall\ \mathbf{a} \in \mathbf{A} \tag{7}$$

and

$$\min_{l=1,\ldots,L} g_l(\mathbf{b}) > 0 \quad \forall\ \mathbf{b} \in \mathbf{B}. \tag{8}$$

Based on PCFs, several competitive classification methods have been developed and some of them have successfully been applied to real-life problems [1, 2, 10–21].

4. The proposed method

In this study, we propose a new method to classify two finite point sets $\mathbf{A}$ and $\mathbf{B}$ by combining the idea of PCFs and SVMs. As we mentioned above, there are different PCF-based classification approaches. In this study, we used k-means based PCF algorithm [11] for the PCF part of the proposed method. In this proposed method, firstly, we chose a class as set $\mathbf{A}$ and the rest as set $\mathbf{B}$. Next, we divided the set $\mathbf{A}$ into k clusters with k-means algorithm and then, we constructed k PCFs by solving k simple linear programming problems. We calculated point-wise minimum of these k PCFs and add this value as a new feature. We repeated this process for all classes. Finally, η features were added to the dataset (training and test sets). This process plays a kernel-like role. After the modification of the dataset, a linear SVM model was applied to solve the new problem. The proposed algorithm is given as follows:

Algorithm:

Step 1. For $i = 1, \cdots, \eta$:

Step 1.1. Set i^{th} class as $\mathbf{A}$ and the rest as $\mathbf{B}$.

Step 2.2. Obtain clusters of the set $\mathbf{A}$ as $\mathbf{A_j}$, $j = 1, \ldots, k$ by applying k-means algorithm.

Step 3.3. For $j = 1, \cdots, k$:
Solve the linear programming problem P_j over the sets $\mathbf{A_j}$ and $\mathbf{B}$; use centroid of the set $\mathbf{A_j}$, as the center $\mathbf{c_j}$; obtain the PCF $\mathbf{g_j}$.

Step 4.4. Extend the training and the test set by a feature. For all points in the test and training set; calculate this feature as point-wise minimum of all PCFs:

$$\min_{j=1,\ldots,k} g_j(.) \tag{9}$$

Step 2. Train the linear SVM method on the extended dataset.

The k-means algorithm [22] is one of the well-known algorithms for solving clustering problems. Since it is easy to implement, fast, and applicable to very large datasets, we preferred to use it in the proposed algorithm. The steps of the k means algorithm are explained as follows.

The k-means Algorithm

Step 1. Choose a seed solution consisting of k centers from set 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.

It should be noted that, clearly, by applying the proposed method, the training and the test set is extended by η features.

4.1. Illustrative example

In order to illustrate the proposed method, first we generated a hypothetical data which cannot be separable with linear structures. These datasets A and B are shown in Figure 4. The data points belonging to sets A and B are represented by red and blue dots, respectively.

According to the proposed approach, all the data points in the datasets are transformed to a new space by using the kernel-like PCFs obtained in the algorithm. By applying Step 1.2 of the proposed algorithm, the centroids and clusters are found as the results of the k-means algorithm. For the $k = 3$ case, PCFs are obtained by solving corresponding problem P. Next, the separating function which is the point-wise minimum of these three PCFs have been calculated as explained in the algorithm's following steps. This separating function and 2d scatter of the data points belonging to sets $\mathbf{A}$ and $\mathbf{B}$ are shown in in Figure 5. In Figure 6, this transformation is illustrated.

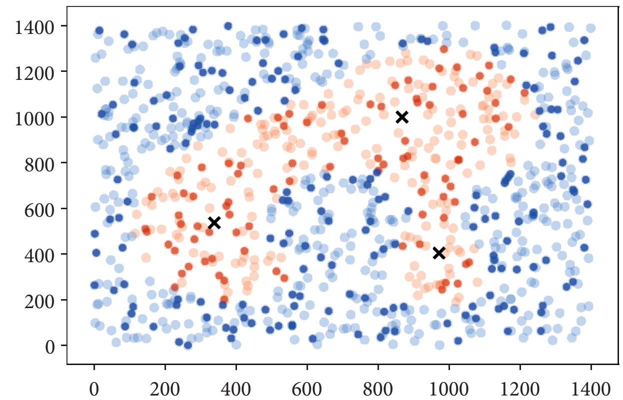

Figure 4. The **A** and **B** data points for illustrative example.

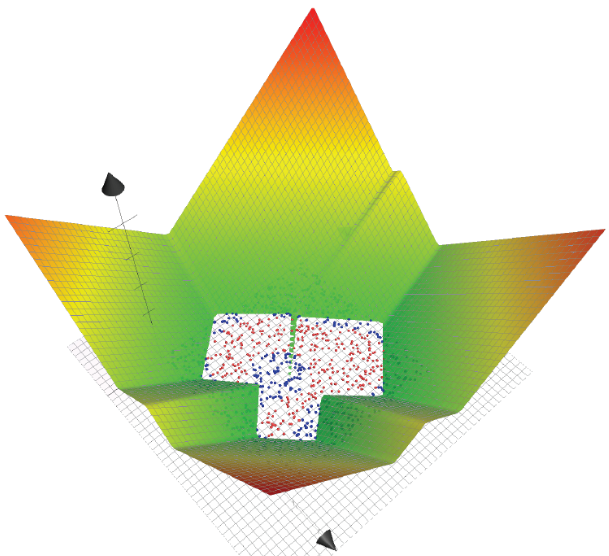

Figure 5. The separating function and 2d scatters of data.

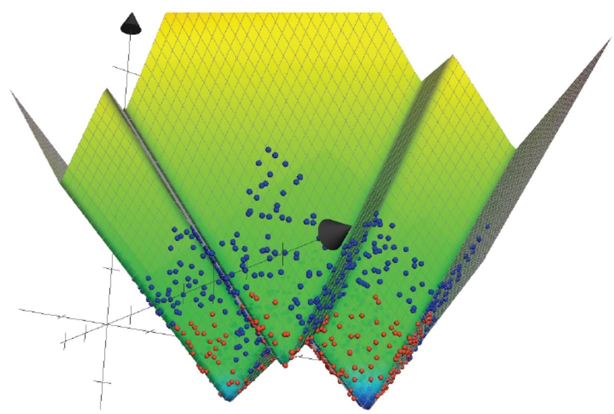

Figure 6. Projection of data points of sets **A** and **B** on separating function.

5. Computational results

The efficiency of the proposed algorithm was tested with publicly available datasets [23]. We used three datasets, namely, Liver, Splice, and Svmguide1. The brief description of the datasets is given in Table 1, where the number of data points for training and test sets, number of classes, and number of attributes for each datasets are included. The datasets given in Table 1 have only real or integer attributes and there is no missing value in them.

The proposed algorithm was implemented in Python 2.7 and MATLAB. The linear programming problem P in Step 1.3 of the proposed algorithm was solved by using Gurobi package[24]. Linear SVM problem in Step 2 was solved with Liblinear package [23].

For comparison of the results, cross-validation procedure was used in the algorithm. In v-fold cross-validation, we first divided the training set into v subsets of equal size. Sequentially one subset was tested using the classifier trained on the remaining $v-1$ subsets. Thus, each instance of the whole training set was predicted once so the cross-validation accuracy is the percentage of data which are correctly classified [9].

Table 1. The brief description of datasets.

	Data sets	Liver	Splice	Svmguide1
	# training data points	145	1000	3089
	# test data points	200	2175	4000
	# attributes	5	60	4
	# classes	2	2	2

Ten-fold cross validation classification accuracy results of the test sets obtained by the proposed algorithm are presented in Table 2. The performance of the proposed method is compared with those of linear and RBF, PUK, and Polynomial kernel SVMs. RBF, PUK, and Polynomial kernel SVMs performances were calculated by using Weka [25]. One can see that the proposed method (PCF + Linear SVM) achieves good classification accuracies. The performance of the proposed method is competitive with RBF, PUK, and Polynomial kernel SVMs. However, the proposed method could not beat the RBF kernel SVM in three tests.

It should be noted that RBF, PUK, and Polynomial kernel SVMs parameters are optimized with a basic grid search. The advantage of the proposed method is that there is no need to optimize any parameters. RBF, PUK, and Polynomial kernel SVM performances strongly depend on parameter selection. However, it can be impossible to optimize kernel function parameters with grid search, especially on big datasets.

Table 2. Comparison of ten-fold cross-validation results

	Liver	Splice	Svmguide1
Linear	54.50	84.37	78.93
Poly.	60.00	84.60	95.60
PUK	60.00	87.08	96.23
RBF	62.50	89.79	96.70
PCF+Linear SVM	61.00	88.14	96.38

6. Conclusion

We proposed a new method that can be used as a kernel-like function for linear SVMs.This method combines PCF and SVM approaches by considering strongest part of these method.

SVMs can generate nonlinear separation surfaces thanks to kernel functions. However, the success of the classifier strongly depends on the selection of the kernel function parameters. In order to obtain the best parameters giving the highest training accuracy, generally, a grid search is applied before training. However, this process can be very time-consuming on big datasets. The proposed method does not need to parameter optimization.

Classification accuracies of the proposed method were compared with linear and RBF, PUK, and Polynomial kernel SVM. We obtained competitive accuracies and it is very promising for the usage the proposed method on big datasets. We are planning to extend computational tests and analysis on big datasets in future research.

References

[1] Gasimov R, Ozturk G. Separation via polyhedral conic functions. Optimization Methods & Software 2006; 21(4): 527-540.

[2] Uylas Sati N. A binary classification approach based on support vector machines via polyhedral conic functions. Celal Bayar University Journal of Science 2016; 12(2): 135-149.

[3] Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995; 20(3): 273-297.

[4] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92. ACM, New York, NY, USA, 1992; 144-152.

[5] Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins C. Text classification using string kernels. Journal of Machine Learning Research 2002; 2: 419-444.

[6] Schölkopf B, Tsuda K, Vert JP. Kernel Methods in Computational Biology. Cambridge, Massachusetts, London, England: MIT Press, 2004.

[7] Gönen M, Alpaydın E. Multiple kernel learning algorithms. Journal of Machine Learning Research 2011; 12: 2211-2268.

[8] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992; 144-152.

[9] Hsu CW, Chang CC, Lin CJ. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[10] Bagirov AM, Ugon J, Webb D, Ozturk G, Kasimbeyli R. A novel piecewise linear classifier based on polyhedral conic and max–min separabilities. Top 2013; 21(1): 3-24.

[11] Ozturk G, Ciftci MT. Clustering based polyhedral conic functions algorithm in classification. Journal of Industrial and Management Optimization 2015; 11(3): 921-932.

[12] Ozturk G, Bagirov AM, Kasimbeyli R. An incremental piecewise linear classifier based on polyhedral conic separation. Machine Learning 2015; 101(1-3, SI): 397-413.

[13] Cimen E, Ozturk G. Arrhythmia classification via k-Means based polyhedral conic functions algorithm. In: Arabnia, HR and Deligiannidis, L and Yang, M, editor, 2016 International Conference on Computational Science & Computational Intelligence (CSCI). Amer Council Sci & Educ, 2016; 798-802. International Conference on Computational Science and Computational Intelligence (CSIC), Las Vegas, NV, USA, 15–17 December, 2016.

[14] Ozturk G, Ceylan G. Max margin polyhedral conic function classifier. In: Arabnia, HR and Deligiannidis, L and Yang, M, editor, 2016 International Conference on Computational Science & Computational Intelligence (CSCI). Amer Council Sci & Educ, 2016; 1395-1396. International Conference on Computational Science and Computational Intelligence (CSIC), Las Vegas, NV, USA, 15–17 December, 2016.

[15] Cimen E, Ozturk G, Gerek O N. Incremental conic functions algorithm for large scale classification problems. Digital Signal Processing 2018; 77: 187-194.

[16] Cimen E, Ozturk G, Gerek ON. Icf: an algorithm for large scale classification with conic functions. SoftwareX 2018; 8: 59-63.

[17] Cevikalp H, Triggs B. Polyhedral conic classifiers for visual object detection and classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017; 4114-4122.

[18] Cimen E. Gesture Recognition with Polyhedral Conic Functions based Classifiers. MSc, Graduate School of Science, Anadolu University, 2013 (in Turkish).

[19] Sati NU. A binary classification algorithm based on polyhedral conic functions. Düzce Üniversitesi Bilim ve Teknoloji Dergisi 2015; 3(1): 152-161.

[20] Sati NU, Ordin B. Application of the polyhedral conic functions method in the text classification and comparative analysis. Scientific Programming 2018; 2018: 1-11.

[21] Sarıbaş H, Çevıkalp H, Kahvecıoğlu S. Car detection in images taken from unmanned aerial vehicles. In: 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018; 1-4.

[22] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. University of California Press, Berkeley, CA, USA. pp. 281-297.

[23] Chang CC, Lin CJ. Libsvm: A library for support vector machines. ACM Trans Intell Syst Technol 2011; 2(3): 1-27.

[24] Gurobi Optimization L. Gurobi optimizer reference manual, 2018.

[25] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 2009; 11(1): 10-18.