A Binary Classification Approach Based on Support Vector Machines via Polyhedral Conic Functions

Nur Uylaş SATI

Muğla Sıtkı Koçman Üniversitesi Bodrum Denizcilik Meslek Yüksek Okulu, Bodrum/Muğla
Tel: +90252 211 5781, nursati@mu.edu.tr

Recieved / Geliş: 18th February (Şubat) 2016
Accepted / Kabul: 13th August (Ağustos) 2016
DOI: http://dx.doi.org/10.18466/cbujos.93628

Abstract

Classification is a frequently used technique of data mining. Binary classification is a type of classification that includes two classes. This problem has a lot of application areas like medical and social sciences, economics, engineering, finance and management, marketing etc. Different mathematical programming approaches of the binary classification have been presented in recent years to support vector machines and polyhedral conic functions. In this paper, a modified algorithm that combines both support vector machines approachment and polyhedral conic functions has been presented. Besides clustering method, a data mining technique, has been added to reduce computational time. Results of numerical experiments on real-world datasets demonstrate that the proposed approach is efficient for solving binary data classification problems. Only one problem arised in some datasets. Because of clustering method's sensitivity on initilization point choice, it can find different local solutions from global ones in some big datasets that have noise or outliers. All results are presented in tables.

Keywords: Classification, Clustering, Mathematical Programming, Polyhedral Conic Functions, Support Vector Machines.

Çokyüzlü Konik Fonksiyonlar ile Destek Vektör Makineleri Tabanlı İkili Sınıflandırma Yaklaşımı

Özet

Sınıflandırma sıkça kullanılan bir veri madenciliği tekniğidir. Bir sınıflandırma çeşidi olan ikili sınıflandırmada iki sınıf bulunur. Bu problemin birçok uygulama alanı vardır; tıp, sosyal bilimler, ekonomi, mühendislik, finans ve yönetim, pazarlama vb. gibi. Son yıllarda ikili sınıflandırma için farklı matematiksel programlama yaklaşımları sunulmuştur. Destek vektör makineleri ve çokyüzlü konik fonksiyonlar bunlardan sadece ikisidir. Makalede bu iki verimli yöntemin kombinasyonuyla oluşturulmuş yeni bir algoritma sunulmuştur. Ayrıca yine bir veri madenciliği yöntemi olan kümeleme metoduda bu algoritmaya eklenerek hesaplama zamanı indirgenmeye çalışılmıştır. Gerçek hayat veri kümeleri üzerinde yapılan sayısal deney sonuçları sunulan yaklaşımın, ikili veri sınıflandırma problem çözümlerinde etkili olduğunu göstermektedir. Sadece bazı veri kümelerinde kümeleme metodunun başlangıç nokta seçimlerindeki hassasiyeti sebebiyle bir problem ortaya çıkmıştır, öyle ki aykırılıklar ve gürültüye sahip büyük veri kümelerinde genel sonuçlardan farklı yerel sonuçlar elde edilmiştir. Tüm sonuç değerleri tablolarda sunulmuştur.

Anahtar Kelimeler: Sınıflandırma, Kümeleme, Matematiksel Programlama, Çokyüzlü Konik Fonksiyonlar, Destek Vektör Makineleri.

1 Introduction

Classification is one of the main techniques of data mining. In classification two groups of labeled datasets (training and test datasets) are dealed, the aim is forming new rules by using training class and examining the effectiveness of rules on test class. Data classification has a lot of applications in various areas like medical and social sciences, economics, engineering, finance and management, marketing etc. [1]. For instance optical character recognition in engineering, predicting the disease by symtomps in medicine, market basket analysis in marketing or fraud detection in banking [2].

There are various methods for binary classification. They are based on machine learning, statistics, neural networks, genetic algorithms, rough and fuzzy set, *k*-nearest neighbor, optimization and mathematical programming etc. [3]. Support vector machines, polyhedral conic functions and clustering methods which are used in our method are based on optimization and mathematical programming.

Binary classification that separates two discrete point sets by finding an appropriate surface in IR^n is based on mathematical programming. Various mathematical programming techniques for binary classification problems were used in the past years. Pattern seperation problem which is a binary classification problem is formulated and solved as a convex programming problem i.e., the minimization of a convex function subject to linear constraints, in [4]. Mangasarian [5] has that main idea: one way to achieve separation is to construct a plane or a nonlinear surface such that one set of patterns lies on one side of the plane or the surface, and the other set of patterns on the other side. A technique for finding such a hyperplane is described by Bennett and Mangasarian [6] and some algorithms based on similar approach are developed in [2, 7, 8,9].

In 1978 Liittschwager and Wang [10], proposed a classification problem that is formulated as a mixed integer programming problem. The solution provides a nonparametric classification statistics which minimizes the expected total cost of misclassification. Also an enumaration algorithm is developed for the special case of class number 2 and it is shown that the performance of the enumeration algorithm is significantly better than Anderson's normal procedure.

In Vapnik [11], a relevant role has also been played by Vapnik-Chervonenkis statistical learning theory which, together with the notion of separation margin, has led to the introduction of the support vector machine (SVM) approach. The aim of Support Vector Machines that is originally developed by Vapnik and co-workers, is to devise a computationally efficient way of learning 'good' hyperplanes in a high dimensional feature space, where by 'good' hyperplanes it is understood that the ones optimising the generalisation bounds [9] . If the problem is nonlinear, instead of trying to fit a nonlinear model in SVMs, kernel functions can be used. Kernel methods in SVM, map the data to high dimensional space where it is easier to classify with linear decision surfaces. The linear model in the new space corresponds to a nonlinear model in original space [12]. This method was used and studied in [13, 14, 15, 16, 17]. In this paper we work in the original space by using a nonlinear model so we dont need kernel functions.

Bagirov, Rubinov and Yearwood reduced the classification problem solving a global optimization problem. A method based on a combination of the cutting angle method and a local search is applied for the solution of the problem. Also this method can be used with an arbitrary number of classes [18, 19].

Astorino and Gaudiso used *h* hyperplanes, generating a convex polyhedron, for separating two finite point sets A and B in the *n*-dimensional space [20]. It is shown that if the intersection of the convex hull of A with B is empty, the two sets can be strictly separated (polyhedral separability). And also an error function which is piecewise linear, but not

neither convex nor concave, was constructed, and descent procedure based on the iterative solution of the LP descent direction finding subproblems was defined. Also the same authors introduced ellipsoidal separation for classification problems [20].

Bagirov described a method called max-min separability that solves the problem by a finite number of hyperplanes generating a piecewise linear function [2]. An error function was described and an algorithm for its minimization was proposed. This method can be considered as generalization of *h* polyhedral separability that was proposed in [20]. It is shown that if the intersection of the A and B sets is empty, then they can be strictly separated by a max–min of linear functions.

In [21] the problem of separating two finite point sets A and B in the *n*-dimensional space was solved by using a special type of polyhedral conic function. An effective finite algorithm for finding a separating function based on iterative solutions of linear programming subproblems is suggested. At each iteration a function whose graph is a polyhedral cone with vertex at a certain point is constructed and the resulting separating function is defined as a point-wise minimum of these functions. In Section 3 PCFs are defined in detail.

In [22] an algorithm for finding piecewise linear boundaries between pattern classes was developed. This algorithm consists of two main stages. In the first stage a polyhedral conic set, introduced in [21] , is used to identify data points which lie on or close to the boundary and in the second stage a piecewise linear boundary, introduced in [2], is computed using only those data points. Piecewise linear boundaries are computed incrementally starting with one hyperplane. Such an approach allows one to significantly reduce the computational effort in many large data sets and an arbitrary number of classes.

In this paper, for the non-separable cases with straight lines and hyperplanes, utilization of polyhedral conic functions in SVMs approachment is suggested. An algorithm that uses SVM approachment with PCFs and also clustering method for decreasing computational time is proposed for binary classification.

In Section 2 of the paper, support vector machines (SVM) studied for both linearly separable and non-separable cases. In Section 3 polyhedral conic functions(PCFs) are explained. Polyhedral conic functions are used in SVM approachment and new algorithms are presented in Section 4 for both separable and non-separable cases. Besides, these algorithms are regenerated by using clustering method for decreasing computational time. Examples are given and illustrations are used for better comprehension. In Section 5, presented algorithm is applied to real-world datasets by using MATLAB, also other classification methods from WEKA is applied to the same datasets for comparison. The numerical results are presented in tables. And finally we conclude the paper in Section 6.

2 Binary Classification with Support Vector Machines

Support vector machines method was defined in 1990 by Vapnik and his friends. This method is based on discriminant- based optimization and finds linear separator parameters by using labeled data sets. It is used by many researchers in various areas [23].

2.1 The Separable Case: The Maximal Margin Classifier

D data set consists of $(x_1, y_1)(x_2, y_2)...(x_n, y_n)$ elements. Here n is the number of elements and $y \in \{+1, -1\}$ stands for the classes, x in case a vector that represents data attribute [9].

As in Figure 2.1 data from two different classes can be separated by a lot of unlike straight lines. In multidimensional space, hyperplanes takes the place of these straight lines.

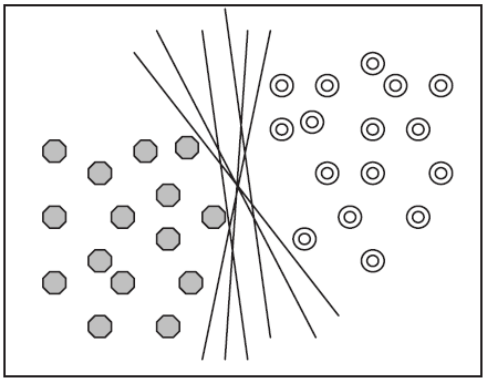

Figure 2.1. Staright lines that separates data defined in R^2

All hyperplanes classify data correctly (See. Figure 2.1) but for a better generalization we want a definite distance between data and hyperplane. This distance is called margin and it is tried to be maximized. The most appropriate separator hyperplane is the one that has the maximized margin. In simple terms, for separation to choose the hyperplanes that have the maximum margin in between is the most appropriate way.

Consider H_2 and H_1 hyperplanes on Figure 2.2. H_0 hyperplane that has the maximum margin between these two hyperplanes, is a linear hyperplane that separates two data classes. This H_0 plane is called "**optimal separating hyperplane".**

H_0 plane is defined below, in terms of points on it :

$$H_0 : W^T X + b = 0$$

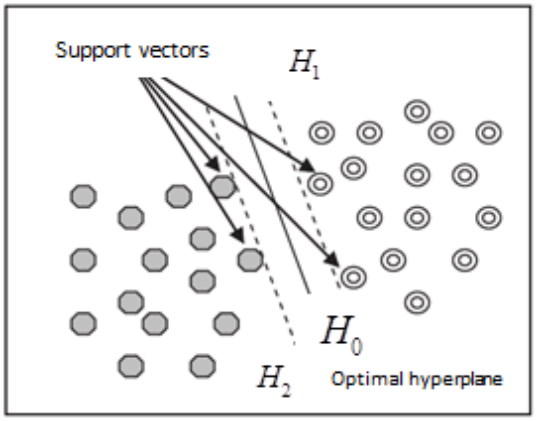

Figure 2.2. Optimal hyperplane

This definition can be written as below:

$$\sum_{i=1}^{n} w_i x_i + b = 0$$

Here W represents weight vector $W = \{w_1, w_2, ..., w_n\}$. n and b are respectively number of attributes and a fixed number.

H_1 hyperplane is expressed as following:

$$H_1 : W^T X + b = 1$$

Likewise H_2 hyperplane is stated as following:

$$H_2 : W^T X + b = -1$$

These inequalities can be expressed as one by combining them as follows

$$y_i (W^T X + b) - 1 \geq 0 \quad \forall i .$$

When we consider H_1 and H_2 hyperplanes, the observations on them are called "**support vectors**".

Here, *d* and *margin* values are calculated as follows .

$$d = \frac{|b|}{\|w\|} - \frac{|b+1|}{\|w\|} = \frac{1}{\|w\|},$$

$$\text{margin} = 2d = \frac{2}{\|w\|}.$$

$\langle .,. \rangle$ and $\|.\|$ consequently represents vector product and two norm.

Maximization of margin means minimization of denominator so this problem is dealed as a minimization problem and defined as follows:

$$\min \frac{\langle w, w \rangle}{2}$$

$$y_i \left(\langle x_i, w \rangle + b \right) - 1 \geq 0.$$

For solution Langrange formulation of the problem is used:

$$L(w,b,\alpha) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle w, x_i \rangle + b \right) - 1 \right]$$

Here $\alpha_i \geq 0, i = 1,...,l$, are Lagrange multipliers, one for each of the inequality constraints [23].

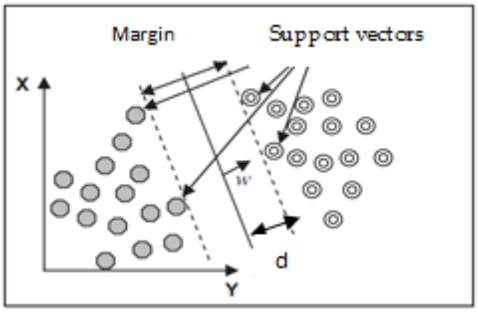

Figure 2.3. *d* and Margin

2.2 Non-Separable Case: The Soft Margin Classifier

In non-separable case of data sets, ξ_i non-negative slack variables that defines misclassifications are added to the optimization model as follows.

$$w^T x_i + b \geq 1 - \xi_i, \quad \text{for } y_i = +1$$

$$w^T x_i + b \leq -1 + \xi_i, \quad \text{for } y_i = -1$$

or,

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad (i = 1,2,..,n), \ \xi_i \geq 0$$

Here x_i data for $\xi_i > 1$, is wrong classified ones that prevent linear separation. Besides x_i data for $0 < \xi_i < 1$ are right classified but takes place in margin area [24].

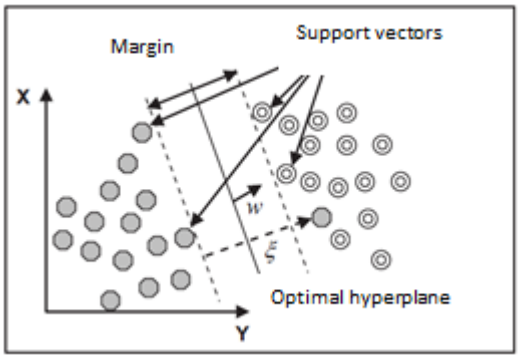

Figure 2.4. Graphical display of misclassification ξ

Finally obtained minimization problem is expressed as follows:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad (i = 1,2,..,n), \ \xi_i \geq 0$$

3 Polyhedral Conic Functions

Polyhedral conic functions are constituted by a linear part adding for l_1 norm expansion to a special class of polyhedral functions that's defined in 2001 by Gasimov. The graph of these functions have sublevel sets that includes mostly 2^n subspace intersection and are polyhedral cones whose vertexes are defined by

$(a,-\gamma)\in R^n\times R$. All these definitions are proofed by Gasimov and Öztürk [21, 25].

$g_{(w,\xi,\gamma,a)}:R^n\to R$ polyhedral conic functions

are defined as follows:

$$g_{(w,\xi,\gamma,a)}:R^n\to R=w'(x-a)+\xi\left\|x-a\right\|_1-\gamma$$

where

$$w,a\in R^n,\xi,\gamma\in R,\quad w'x=w_1x_1+...+w_nx_n\ ,$$

$$\left\|x\right\|_1=\left|x_1\right|+...+\left|x_n\right|\ .$$

Sublevel sets of polyhedral conic functions are polyhedrons for $\alpha\in R$ as follows:

$$S_\alpha=\left\{x\in R^n:g(x)\le\alpha\right\}.$$

In accordance with these definitions if the graph of $g_{(w,\xi,\gamma,a)}:R^n\to R$ function is a cone and for $\alpha\in R$ all sublevel sets are polyhedrons, this g function is called polyhedral conic functions [12].

In figure 3.1, we show graphs of polyhedral conic functions in R^2 for a better understanding. These functions are constituted by assigning different values to w,ξ,γ and a parameters. MATLAB is used for plotting.

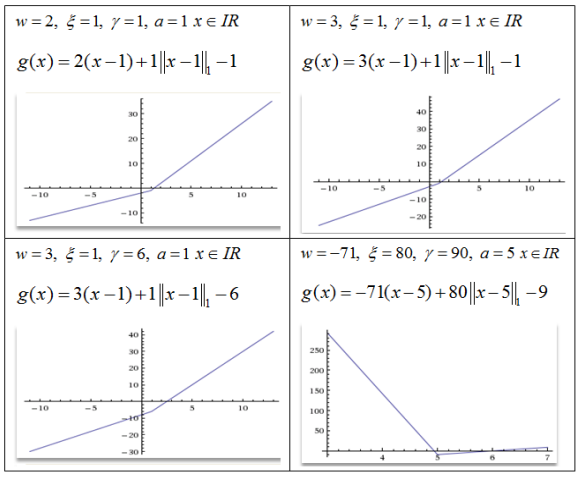

Figure 3.1 Polyhedral Conic Functions in R²

Separation with polyhedral conic functions is studied by Gasimov and Öztürk [21] in 2006. To find a separating function, an effective finite algorithm is developed.

4 Support Vector Machines with Polyhedral Conic Functions

In this section, a new separation method is developed by using polyhedral conic functions instead of straight lines or hyperplanes in support vector machines approachment. In 4.1 an algorithm for separable cases with polyhedral conic functions is constituted and in 4.2 for non-separable cases a misclassification function is defined and an algorithm that minimizes this function is presented. In 4.3 examples are given for a better understanding and also the existing algorithm is changed by using clustering method in vertex choosing part to increase the efficiency.

4.1. Separable case with polyhedral conic functions

Let A and B are two given sets in R^n:

$$A=\left\{a^i\in R^n:i\in I\right\},B=\left\{b^j\in R^n:j\in J\right\}$$

where $i=\{1,..,m\}$, $j=\{1,..,p\}$.

Lemma 4.1: A and B sets are separable with polyhedral conic functions if there is at least one $g(x)=g_{(w,\xi,\gamma,a)}(x)$ polyhedral conic function as follows:

$$\begin{aligned}g(a^i)&\le 0\quad\forall a^i\in A\\ g(b^j)&>0\quad\forall b^j\in B\end{aligned}\ .$$

As can be seen in Figure 4.1 datasets for n=1, can be separated by various polyhedral conic functions that have $a,w\in R^n$, $\xi\in R$ fixed parameters and different $\gamma\in[1,\infty)$ parameters (hence different $(a,-\gamma)\in R^n\times R$ vertex point). The most proper way to separate these datasets

is to choose the ones (PCFs) that has the maximum margin.

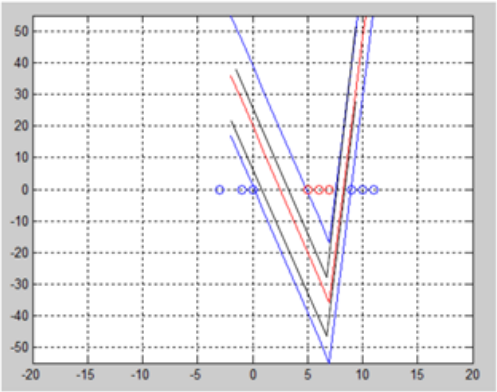

Figure 4.1. two-classed datasets and ve PCFs

$g_1(x)$ and $g_2(x)$ PCFs in Figure 4.3 are the ones that have the maximum margin and $g(x)$ between these two PCFs is the polyhedral conic function that separates these two-classed datasets. This $g(x)$ polyhedral conic function is called "**optimal separating polyhedral conic function"**.

When we consider $g_1(x)$ and $g_2(x)$ polyhedral conics, the observations on these conics are called **"support vectors"**. In a word; $g_1(x)=0, x\in A$ and $g_2(y)=0, y\in B$ vectors are support vectors.

Considered polyhedral conic functions are stated as follows:

$$g_1(x)=w(x-a)+\xi\|x-a\|_1-\gamma_1$$

$$g_2(x)=w(x-a)+\xi\|x-a\|_1-\gamma_2$$

$$g(x)=w(x-a)+\xi\|x-a\|_1-\gamma$$

The Euclidian distance between two vertex points $(a,-\gamma_1)$, $(a,-\gamma_2)$ of these PCFs for $\gamma_2\geq\gamma_1$ can be computed as follows:

$$\sqrt{(a-a)^2+(\gamma_2-\gamma_1)^2}=|\gamma_2-\gamma_1|=\gamma_2-\gamma_1$$

Finally the obtained minimization problem model is expressed below:

$$(P)\quad \min-(\gamma_2-\gamma_1)$$

$$w(a^i-a)+\xi\|a^i-a\|_1-\gamma_1\leq-1,\quad i=1,..,m,$$

$$w(b^j-a)+\xi\|b^j-a\|_1-\gamma_2\geq 1,\quad j=1,..,p,$$

$$\gamma_2-\gamma_1\geq 0,$$

$$w\in R^n,\xi\in R,\gamma_{1,2}\geq 1,$$

"**Optimal separating polyhedral conic function**" that's obtained by solving (P) minimization problem is defined as follows:

$$g(x)=w(x-a)+\xi\|x-a\|_1-(\frac{\gamma_1+\gamma_2}{2}).$$

At the initilization step of the algorithm for getting efficient results, at each iteration l, the problem (P_l) is solved and the numbers of elements l_i from A_l separated from B are found. Then a_l is defined as $a_l=a_{l_0}$ where $l_0=\max\{l_i: i\in I\}$ [12].

Algorithm 1: SVM-PCF 1

Let A and B are two sets defined in $\mathbb{R}^n$.

$$A=\{a^i\in R^n: i\in I\}, B=\{b^j\in R^n: j\in J\}$$

where $i=\{1,..,m\}$, $j=\{1,..,p\}$.

Initilization step: For $\forall a_i\in A$ solve (P) subproblem.

The number of points that's included in A not in B $=l_i$, $a=a_{l_0}$, $l_0=\max\{l_i: i\in I\}$

Step 1: Solve P subproblem.

$$(P)\quad \min-(\gamma_2-\gamma_1)$$

$$w(a^i-a)+\xi\|a^i-a\|_1-\gamma_1\leq-1,\quad i=1,..,m,$$

$$w(b^j-a)+\xi\|b^j-a\|_1-\gamma_2\geq 1,\quad j=1,..,p,$$

$$\gamma_2 - \gamma_1 \geq 0,$$

$$w \in R^n, \xi \in R, \gamma_{1,2} \geq 1,$$

$w, \xi, \gamma_1, \gamma_2$, is a solution of *P*.

and $\gamma = \frac{\gamma_2 + \gamma_1}{2}$.

Step 2: Define $g(x)$ function separates *A* and *B* sets as follows:

$$g(x) = g_{(w,\xi,\gamma,a)}(x).$$

4.2 Non-separable case with polyhedral conic functions

In non-separable case of datasets, ξ_i non-negative slack variables that defines misclassifications are added to the optimization model. Also the following designed Algorithm 2 can be applied to separable cases, in this case the misclassification values for $i = 1,..m,\ j = 1,..,p$ is $\varepsilon_i, \varepsilon_j = 0$.

$$w(a^i - a) + \xi \left\| a^i - a \right\|_1 - \gamma_1 \leq -1 + \varepsilon_i, \quad i = 1,..,m,$$
$$w(b^j - a) + \xi \left\| b^j - a \right\|_1 - \gamma_2 \geq 1 - \varepsilon_j, \quad j = 1,..,p,$$

Here a^i and b^j indexed data for $\varepsilon_i, \varepsilon_j > 0$ are respectively on the reverse side of g_1 and g_2 polyhedral conic functions namely wrong classified data.

Algorithm 2: SVM-PCF 2

Let A and B are two sets defined in R^n.

$$A = \left\{ a^i \in R^n : i \in I \right\}, B = \left\{ b^j \in R^n : j \in J \right\}$$

where $i = \{1,..,m\}, \quad j = \{1,..,p\}$.

Initilization step: For $\forall a_i \in A$ solve (P) subproblem.

The number of points that's included in A not in B = l_i

$$a = a_{l_0},\ l_0 = \max \left\{ l_i : i \in I \right\}$$

Step 1: Solve P subproblem.

$$(P) \quad \min(-(\gamma_2 - \gamma_1) + \sum_{i=1}^{m} \varepsilon_i + \sum_{j=1}^{p} \varepsilon_j)$$

$$w(a^i - a) + \xi \left\| a^i - a \right\|_1 - \gamma_1 \leq -1 + \varepsilon_i, \quad i = 1,..,m,$$

$$w(b^j - a) + \xi \left\| b^j - a \right\|_1 - \gamma_2 \geq 1 - \varepsilon_j, \quad j = 1,..,p,$$

$$\gamma_2 - \gamma_1 \geq 0,$$

$$\varepsilon_{i,j} \geq 0, w \in R^n, \xi \in R, \gamma_{1,2} \geq 1,$$

$w, \xi, \gamma_1, \gamma_2, \varepsilon_i, \varepsilon_j$ for *i=1,..,m, j=1,…,p* , is a solution of *(P)*.

and $\gamma = \frac{\gamma_2 + \gamma_1}{2}$.

Step 2: Define $g(x)$ function separates *A* and *B* sets as follows:

$$g(x) = g_{(w,\xi,\gamma,a)}(x).$$

Example 4.1: Consider $A = \{5, 6, 7\}$ and $B = \{-3, -1, 0, 9, 10, 11\}$ sets defined in R. In Figure 4.2 *A* and B points are respectively stated by red and blue and as is seen these *A* and *B* points cannot be separated linearly.

Following results are obtained by applying SVM-PCF 2 algorithm to A and B sets on MATLAB.

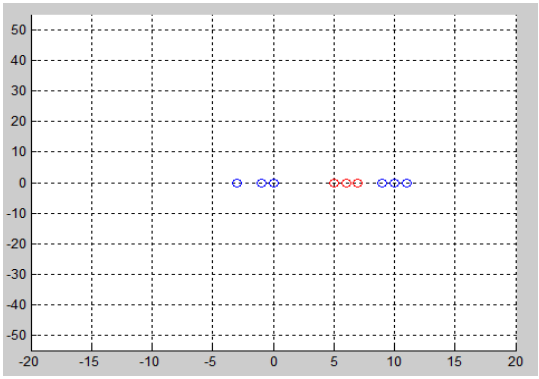

Figure 4.2: Datasets in R^2

The algorithm chooses " *a=7* " point as vertex point of PCF.

$w=10, \xi=18, \gamma_1=17, \gamma_2=55$ values are obtained by solving (*P*) minimization problem. The specified polyhedral conic function whose S_0 sublevel set includes the points of A set, $a_i \in S_0, g_1(a_i) \le 0$, is defined as follows:

$$g_1(x)=10(x-7)+18\|x-7\|-17 .$$

Likewise, the specified polyhedral conic function whose S_0 sublevel set excludes the points of B set, $b_j \in S_0, g_2(b_j) \ge 0$, is defined as follows:

$$g_2(x)=10(x-7)+18\|x-7\|-55 .$$

And finally obtained **"Optimal separator polyhedral conic function"** is calculated as follows:

$$g(x)=10(x-7)+18\|x-7\|-36 .$$

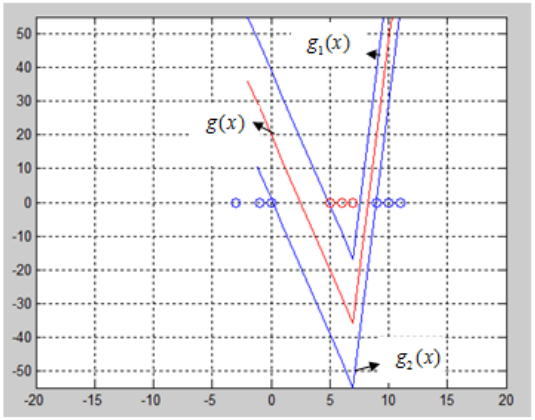

Figure 4.3. Obtained PCFs in R^2

As is seen from Figure 4.3 obtained *optimal separator polyhedral conic function*'s S_0 sublevel set includes points of A and excludes points of B and accordingly following inequalities are ensured.

$$g(a^i) \le 0 \quad \forall a^i \in A$$
$$g(b^j) > 0 \quad \forall b^j \in B$$

Example 4.2: Consider $D=\{(-7,1),(2,-2),(-5,3),(1,1)\}$ and $G=\{(8,1),(5,1),(8,-5),(-1,-5)\}$ datasets in R^2.

In figure 4.4 *D* and *G* points are respectively stated by red and blue and as is seen these *D* and *G* points cannot be separated linearly. Obtained results after applying SVM-PCF 2 is as follows:

$$w=(1,-1), \xi=1, \gamma_1=1, \gamma_2=5, \gamma=\frac{5+1}{2}=3$$

And obtained optimal separator *g(x)* PCF with these parameters is shown in Figure 4.5.

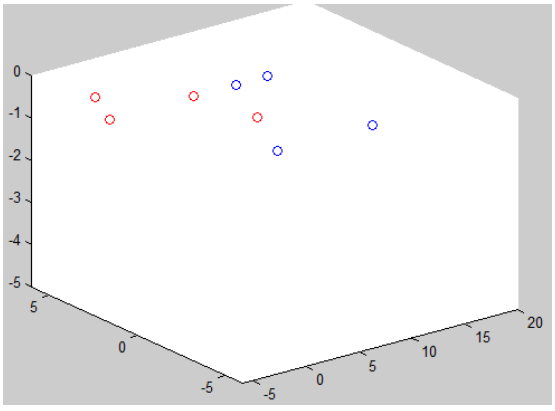

Figure 4.4. Datasets in R^3

When we add (-5,2) point to *G* set acquired dataset notation and the graphical notation of *g(x)* after applying the algorithm SVM-PCF 2 is

presented in Figure 4.6. Obtained parameter values are expressed as follows:

$a=(2,-2), w=(0.33,-0.33),\ \xi=0.33, \gamma_1=1,$

$\gamma_2=1, \gamma=1$.

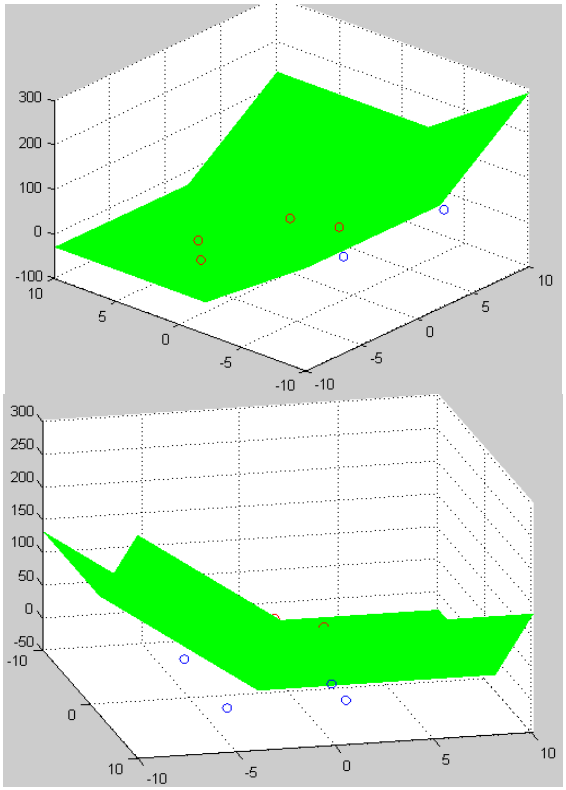

Figure 4.5. Optimal separator PCF in R^3

$\varepsilon_5 =$ misclassification value of b^5 point $= 2.0$

accuracy=88.8

$$g(x)=(0.33,-0.33)\times(x-(2,-2))$$

$$+0.33\|x-(2,-2)\|_1-1$$

Consequently, in this section for the non separable cases in SVM with straight lines and hyperplanes, a new method (SVM-PCF) is developed by using polyhedral conic functions and as is seen from the results, we succeed in separation.

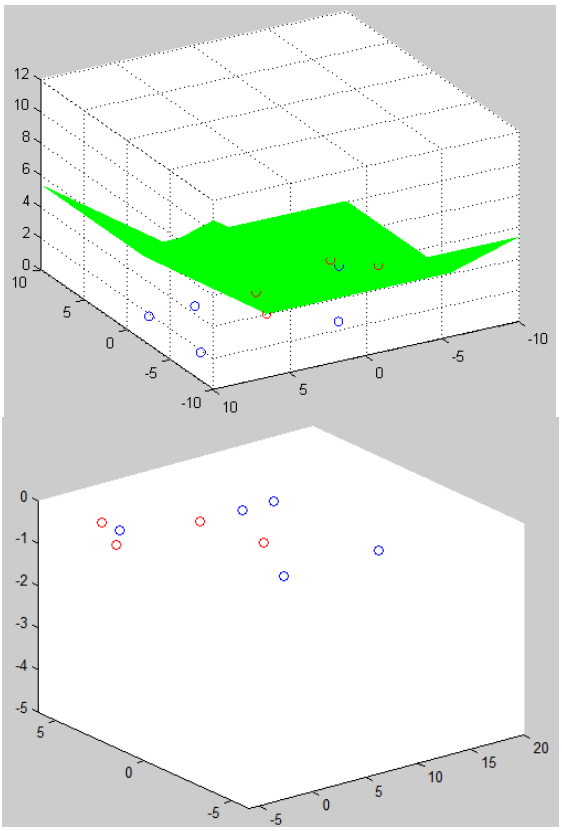

Figure 4.6. Datasets and optimal separator PCF in R^3

4.3 Clustering in SVM-PCF Algorithms

In the designed algorithms the computational time is mostly depends on proper vertex finding because (*P*) subproblem is solved for every point and the most proper one is used as vertex of the cone. This chose increases the accuracy value of the algorithm but the computational time should not be ruled out. For a proper solution, clustering method one of data mining methods, is used for finding vertex point. The *k*-means algorithm, one of clustering method is used in the designed SVM-PCF algorithm [27, 28]. In *k*-means algorithm, for the initilization, a solution consisting of *k* centers is chosen then data points to its nearest center are allocated and *k*-partition of A is obtained. For new partitions centers are recomputed and these processes continue until no more data points change clusters [26].

This method is used in the initilization part of the SVM-PCF algorithm and designed algorithm is expressed as follows:

Algorithm 6: SVM-PCF-CLSTRNG

Let A and B are two sets defined in R^n .

$$A=\left\{a^i \in \square^n : i \in I\right\}, B=\left\{b^j \in \square^n : j \in J\right\}$$

where $i=\{1,..,m\}$, $j=\{1,..,p\}$.

Initilization step: Apply clustering algorithm on set of *A*. Let *s* be the number of clusters and *k=1*

For every cluster center point, $a=a_s$, *for s=1,2,..,k* solve (*P*) subproblem.

l_s = the number of points included in A not in B for ,s=1,2,…,k,

$a=a_{l_0}$, $l_0=\max\{l_s : s=1,2,...,k\}$

Step 1: Solve P subproblem .

$$(P)\quad \min(-(\gamma_2-\gamma_1)+\sum_{i=1}^{m}\varepsilon_i+\sum_{j=1}^{p}\varepsilon_j)$$

$$w(a^i-a)+\xi\left\|a^i-a\right\|_1-\gamma_1\le -1+\varepsilon_i,\quad i=1,..,m,$$

$$w(b^j-a)+\xi\left\|b^j-a\right\|_1-\gamma_2\ge 1-\varepsilon_j,\quad j=1,..,p,$$

$$\gamma_2-\gamma_1\ge 0,$$

$$\varepsilon_{i,j}\ge 0, w\in R^n, \xi\in R, \gamma_{1,2}\ge 1,$$

$w,\xi,\gamma_1,\gamma_2,\varepsilon_i,\varepsilon_j$ for *i=1,..,m, j=1,…,p* , is a solution of *(P)*.

and $\gamma=\dfrac{\gamma_2+\gamma_1}{2}$.

Step 2: Define *g(x)* function separates *A* and *B* sets as follows:

$$g(x)=g_{(w,\xi,\gamma,a)}(x).$$

5 Numerical Experiments

In this section to validate the performance of the proposed algorithm numerical implemantations are performed on the two-classed real-world datasets taken from UCI (UC Irvine Machine Learning Repository) [29]. The description of datasets is given in Table 5.1.

In Table 5.2 the results of SVM-PCF 2 and SVM-PCF-CLSTRNG algorithms are compared with regard to accuracy and computational time. The accuracy is constructed as follows:

ccp= the number of correct classified points

tp= the number of training points

$$\text{accuracy}=\frac{100\times ccp}{tp}$$

In Table 5.3 designed algorithm SVM-PCF-CLSTRNG is compared with the other classification algorithms by using WEKA, in terms of 10-fold cross validation. In [30], 10-fold cross validation is explained. In simple terms, it evaluates the performance of learning algorithms.

Table 5.1 The brief description of datasets

Datasets	Number of Instances	Number of Attributes
Diabetes	768	8
Ionosphere	351	34
Liver	345	6
WBCD	683	9
Heart	297	13
WBCP	194	32
Connectionist Bench	208	60
Haberman	306	3

Table 5.2 The results of implemantations with SVM-PCF and SVM_PCF_CLSTRNG

	SVM-PCF		SVM-PCF -CLSTRNG	
Datsets	Accuracy %	Time Sec.	Accuracy %	Time Sec.
Diabetes	72.73	368	72.39	11.80
Ionosphere	75.6	656	84.33	10.19
Liver	64.95	296	64.63	5.43
WBCD	93.16	300	87.42	8.18

Heart	72.5	180	81.66	2.86
WBCP	64.9	130	76.80	2.11
Connectionist Bench	80.76	128	75.96	2.78
Haberman	74.18	166	74.18	3.94

Table 5.3 The results of LIBSVM and SVM-PCF-CLSTRNG's 10-fold cross validation

	LIBSVM	IBI	Classification via clustering	LWL	SVM-PCF-CLSTRNG
Datasets	10-fold cross validation %	10-fold cross validation %	10-fold cross validation %	10-fold cross validation %	10-fold cross validation %
Diabetes	93.44	70.18	64.83	71.22	70.73
Ionosphere	65.10	86.32	70.94	82.33	80.72
Liver	59.42	62.89	53.62	59.13	64.99
WBCD	66.38	95.13	95.70	90.12	85.39
Heart	55.92	75.18	77.03	71.85	77.17
WBCP	76.26	69.19	59.09	76.26	76.68
Connectionist Bench	65.86	86.53	54.32	73.55	70.62
Haberman	73.52	65.68	48.36	72.87	74.21

6 Results and Discussion

As it is seen from the results in Table 5.2, clustering method is very useful to decrease the computational time of the algorithm, because in SVM-PCF-CLSTRNG algorithm (P) subproblem is solved just for the obtained cluster points not for all the points of A. Beside the utilization of clustering method in computational time, the change of the accuracy results should be considered.

The designed SVM-PCF-CLSTRNG algorithm's 10-fold cross validation results, shown in Table 5.3, are better than the others for most of the datasets. Obtaining worse results in the rest of the datasets is due to the clustering method. "*k* " value that's specified for *k*-means algorithm can change the generalization results due to the overfitting or underfitting. For all the datasets, a fixed number, *k*=20, is used . The method is so sensitive in choosing initialization points that, it can find local solutions different from global ones in datasets that have noise or outliers.

Separation with SVMs and PCFs approachments were both studied recently for multiclass classification [27, 31]. By the help of these studies SVM-PCF-CLSTRNG can be developed to multiclass classification problems for future work.

In this paper an algorithm that uses SVM approachment with PCFs and Clustering method is proposed for binary classification. For the non-separable cases with straight lines and hyperplanes, we aim to find a nonlinear decision surface by the utilization of polyhedral conic functions. For a better comprehension, examples are given and figures are used. Designed algorithms are applied to real-world datasets and comparisons are made with other classification algorithms, taken from WEKA. Results are shown in tables and for implemantations WEKA and MATLAB softwares are used.

7. References

[1] Martin, T. P.; Baldwin, J. F. Intelligent Systems and Soft Computing: Prospects, Tools and Applications, 2000; pp. 161-187.

[2] Bagirov, A.M.; Ugon, J. Supervised Data Classification via Max-Min Separability, Continous Optimization,Applied Optimization, 2005, 99: 175-207.

[3] Michie, D.; Spiegelhalter, D.J. Machine Learning, Neural and Statistical Classification 1994.

[4] Rosen, J.B. Pattern separation by convex programming, 1963, Stanford Univ. Calif. Applied Mathematics and Statistics Labs.

[5] Mangasarian, O.L. Linear and nonlinear separation of patterns by linear programming, Operations Research, 1965, 13/3: 444-452.

[6] Bennett, K.P.; Mangasarian, O.L. Robust linear programming discrimination of two linearly inseparable sets, Optimization methods and software, 1992, 1/1: 23-34.

[7] Astorino, A.; Gaudioso, M. Polyhedral Separability through Successive LP, Journal of Optimization Theory and Applications, 2002, 112/2: 265-293.

[8] Bagirov, A.M. Max Min Separability, Optimization Methods and Software, 2005; 20/ 2-3: 277-296.

[9] Cristianini, N.; Taylor J.S. An Introduction to Support Vector Machines and Kernel-Based Learning Methods, Cambridge University Press, 2000; 3-23.

[10] Liittschwager, J.M.; Wang, C. Management Science , 1978; 24/ 14 :1515-1525.

[11] Vapnik, V. The Nature of Statistical Learning Theory, Springer Verlag N.Y., 1995; 189pp.

[12] Alpaydın, E. Introduction To Machine Learning. The MIT Press Cambridge, 2010; Massachusetts London, England.

[13] Vapnik, V.; Golowich, S. E.; Smola A. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing, Neural Information Processing Systems, 1997; vol 9. MIT Press, Cambridge, MA.

[14] Schölkopf, B. Support Vector Learning, R. Oldenbourg Verlag, 1997; Munich.

[15] Baudat, G.; Anouar, F. Kernel-based Methods and Function Approximation , Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on (Volume:2).

[16] Baudat, G.; Anouar, F. Generalized Discriminant Analysis Using a Kernel , Journal Neural Computation archive Volume 12 Issue 10, October 2000; Pages 2385 - 2404 (2)

[17] Mika, S.; Rätsch, G.; Weston J.; Schölkopf, B.; Müller, K. R. Fisher Discriminant Analysis with Kernels, Proc. IEEE Neural Networks for Signal Processing Workshop, 1999; NNSP.

[18] Bagirov, A.; Rubinov, A.; Yearwood, J. Using global optimization to improve classification for medical diagnosis and prognosis, Top Health Inf. Manage,Aug., 2001; 22/1: 65-74. 7.

[19] Bagirov, A.M.; Rubinov, A.M.; Yearwood, J.V. A Global Optimization Approach to Classification, Optimization and Engineering, 2002; 3/2: 129-155.

[20] Astorino, A.; Gaudioso, M. Ellipsoidal separation for classification problems Optimization Methods and Software, 2005; 20/2-3 : 267-276.

[21] Gasimov, R.N.; Öztürk, G. Separation via polyhedral conic functions, Optimization Methods and Software, 2006; 21/ 4 :527-540.

[22] Bagirov, A.M.; Ugon, J.; Webb, D.; Öztürk, G.; Kasımbeyli, R. A novel piecewise linear classifier based on polyhedral conic and max-min separabilities, 2011; TOP, DOI : 10.1007/s11750-011-0241-5.

[23] Mammone, A.; Turchi, M.; Cristianini, N. Support Vector Machines, Wires: Wiley's Interdisciplinary Reviews in Computational Statistics, June, 2009; 1:283-289.

[24] Bennett, K.P.; Demiriz, A. Semi-supervised support vector machines, In D.A. Cohn M. S. Kearns S. A. Solla. Editor. Cambridge. MA, Advances in Neural Information Processing System, 1998; 10, 368-374.

[25] Gasimov, R.N. Characterization of the Benson Proper Efficiency and Scalarization in Nonconvex Vector Optimization, Multicriteria Decision Making in the new Millenium, 2001; 507: 189-198.

[26] Bagirov, A.M.; Mardaneh, K. Modified global k-means algorithm for clustering in gene expression data sets, WISB '06 Proceedings of the 2006 workshop on Intelligent systems for bioinformatics, 2006; 73: 23-28.

[27] Öztürk, G.; Çiftçi, M. Clustering Based Polyhedral Conic Functions Algorithm in Classification, Journal of Industrial and Managemant Optimization doi:10.3934/jimo.2015.11.921 Volume 11, Number 3, 2015; July.

[28] Satı, U.N. A Binary Classification Algorithm Based On Polyhedral Conic Functions, Düzce University Journal of Science and Technology, 2015; 3, 152-161.

[29] Murphy, P.M.; Aha, D.W. 1992; UCI repository of machine learning databases. Technical report, Department of Information and Computer Science, University of California, Irvine, available online at: www.ics.uci.edu/mlearn/MLRepository.html.

[30] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection, 1995; International Joint Conference on Artificial Intelligence.

[31] Wang. Z.; Xue, X. Support Vector Machines Applications, Springer International Publishing, Switzerland, 2014; Chapter 2 Multi-Class Support Vector Machine.