Düzce University
Journal of Science & Technology

Research Article

A Binary Classification Algorithm Based on Polyhedral Conic Functions

Nur Uylaş SATI*

Department of Mathematics, Faculty of Science, Ege University, İzmir, TURKEY
** Corresponding author's e-mail address: nuruylas@gmail.com*

ABSTRACT

Data classification is one of the main techniques of data mining. Different mathematical programming approaches of the data classification were presented in recent years. A technique that uses polyhedral conic functions (PCF) is an effective method for data classification. We present a modified classification algorithm based on PCF functions. Results of numerical experiments on real-world and synthetic data sets demonstrate that the proposed approach is efficient for solving binary data classification problems.

Keywords: *Mathematical Programming, Polyhedral Conic Functions, Classification, Clustering*.

Çokyüzlü Konik Fonksiyonlar Temelli Bir İkili Sınıflandırma Algoritması

ÖZET

Veri sınıflandırma, veri madenciliğinin önemli tekniklerinden birisidir. Son yıllarda veri sınıflandırması için farklı matematiksel programlama yaklaşımları sunulmuştur. Çokyüzlü konik fonksiyonları kullanan bir teknik veri sınıflandırması için efektif bir yöntem olmuştur. Bu çalışmada çokyüzlü konik fonksiyonları temel alan geliştirilmiş bir sınıflandırma algoritması sunulmuştur. Gerçek hayat ve sentetik veri kümeleri üzerinde yapılan sayısal deney sonuçları göstermektedir ki sunulan yaklaşım ikili veri sınıflandırma problemlerinin çözümünde etkili olmuştur.

Anahtar Kelimeler: *Matematiksel Programlama, Çokyüzlü Konik Fonksiyonlar, Sınıflandırma, Kümeleme.*

Received: 23/12/2014, Revised: 07/01/2015, Accepted: 07/01/2015

I. INTRODUCTION

THE supervised data classification uses data whose classes are known. These data sets are called training sets. The aim of supervised data classification is to define rules on this training set. By using these rules making efficient data classification is expected. The efficiency of the found rule is examined on the test set. Supervised data classification applications can be met up in every area that involves data mining such as business, medicine, engineering etc.

Binary classification problem consists of finding an appropriate surface in IRn separating two discrete point sets and it is particularly based on mathematical programming. Several mathematical programming techniques for binary classification problems were used in [2,3,8,7,9,12]. Some of these techniques are mentioned in the next section.

In this paper an algorithm based on the PCF functions is presented. Firstly we aim to find efficient vertices of cones thus we can develop the algorithm performance and secondly to prevent overfitting because of the difference between training and testing accuricies. In accordance with this purpose the algorithm based on PCF is reformulated, new constraints and methods are added. Numerical experiments have been carried out. In conclusion the results show that the proposed method improved the approach based on PCF for binary data classification.

The rest of the paper is organized as follows: A brief description of binary classification is given in section 2. In section 3, polyhedral conic functions and basic properties of them are given. Besides the PCF algorithm is expressed. In section 4 a new formulation for binary data clasification model is given and an algorithm based on this model is presented. The results of numerical experiments are given in section 5 and finally section 6 concludes the paper.

II. METHOD

A. BINARY CLASSIFICATION

Binary classification problem can be stated as follows: A random couple , where is called the feature vector is given and is called the label. The goal is to learn a classifier, i.e., a mapping that separates the vectors with reference to labels [7]. Verbally, it can be defined as a problem of obtaining a criterion for distinguishing between the elements of two disjoint sets of patterns [13].

Many algorithms have been proposed and studied to solve this problem [2, 3, 5, 6, 7, 8, 9, 12, 13, 14, 15] in the last decades. Most frequently used ones are based on linear, polyhedral and max-min separation.

In paper [14] pattern separation problem which is a binary separation problem is solved as a convex programming problem. In [13] the same idea is used and to achieve separation a plane or a non linear surface, such that one set of patterns lies on one side of the plane and the other set of patterns on the other side, is constructed. In [8] Bennett and Mangasarian presented a method to find such a hyperplane. This method is based on linear separability. In linear separability the convex hulls of the two sets do not intersect. If the intersection is not empty a hyperplane can be constructed by letting some misclassification or nonlinear separating surfaces can be looked for.

The concept of polyhedral separability was introduced in [2]. Astorino and Gaudiso used h hyperplanes, that configures a convex polyhedron, for binary classification. They introduced an error function which is piecewise linear but not convex nor concave.

In [3] Bagirov described max-min separability that is a generalization of h-polyhedral separability. It solves the problem by finite number of hyperplanes that constructs piecewise linear function. And it is proved that if the intersection of two sets is empty they can be strictly separated by a max-min of linear functions .

In this paper PCF, a method that is defined in [9], is used for binary classification. In the next section this method will be analyzed.

B. POLYHEDRAL CONIC FUNCTIONS(PCF)

Polyhedral conic functions (PCFs) have recently been proposed to separate two disjoint point sets in IR^n [9] . In [9] Definition 1 and Lemma 1 quoted below are given and proofed.

Definition 1: A function $g : IR^n \times IR$ is called polyhedral conic if its graph is a cone and all its level sets, $S_\alpha = \{x \in IR^n : g(x) \le \alpha\}, \alpha \in IR$ are polyhedrons.

Given $w, a \in IR^n, \xi, \gamma \in IR, w'x = w_1x_1 + ... + w_nx_n$ is a scalar productof w and x, $\|x\|_1 = |x_1| + ... + |x_n|$ is a l_1 norm of the vector $x \in IR^n$, a polyhedral conic function $g_{(w,\xi,\gamma,a)} : IR^n \to IR$ defined as

$$g_{(w,\xi,\gamma,a)} : IR^n \to IR = w'(x-a) + \xi \|x-a\|_1 - \gamma \tag{1}$$

Lemma 1: A graph of the function $g_{(w,\xi,\gamma,a)}$ defined in (1) is a polyhedral cone with a vertex at $(a, -\gamma) \in IR^n \times IR$. This cone is called a polyhedral conic set and *a* its center.

Let *A* and *B* be given sets containing *m* and *p n*-dimensional vectors, respectively:

$A = \{a^i \in R^n, i \in I\}$, $B = \{b^j \in R^n, j \in J\}$ where $I = \{1, ..., m\}, J = \{1, ..., p\}$.

An algorithm generating a polyhedral conic separating function, called a PCF algorithm, proceeds as follows [9]:

Algorithm 1. PCF Algorithm for binary data classification.

Step 0.(Initialization step) l=1, $I_l = I, A_l = A$ and go to *Step 1*.

Step 1. Let a^l be an arbitrary point of A_l. Solve subproblem P_l.

$$(P_l) \quad \min\left(\frac{y' e_{|I_l|}}{|I_l|}\right) \tag{2}$$

$$w^{'}(a^{i}-a^{l})+\xi\left\|a^{i}-a^{l}\right\|_{1}-\gamma+1\le y_{i},\quad \forall i\in I_{l}, \tag{3}$$

$$-w^{'}(b^{j}-a^{l})-\xi\left\|b^{j}-a^{l}\right\|_{1}+\gamma+1\le 0,\quad \forall j\in J, \tag{4}$$

$$y=(y_{1},...,y_{m})\in R_{+}^{m},w\in R^{n},\xi\in R,\gamma\ge 1 \tag{5}$$

Let $w^{l},\xi^{l},\gamma^{l},y^{l}$ be a solution of (P_{l}) and let

$$g_{l}(x)=g_{(w^{l},\xi^{l},\gamma^{l},a^{l})}(x) \tag{6}$$

and go to *Step 2*.

Step 2. Let $I_{l+1}=\left\{i\in I_{l}:g_{l}(a^{i})+1>0\right\}$, $A_{l+1}=\left\{a^{i}\in A_{l}:i\in I_{l+1}\right\}$, $l=l+1$ and if $A_{l}\neq\varnothing$

go to *Step 1*.

Step 3. Define the function *g(x)* (separating the sets *A* and *B*) as

$$g(x)=\min_{l}g_{l}(x) \tag{7}$$

and stop.

In this algorithm the number of iterations causes efficiency decreasing and it is strongly depends on the place of the vertex of polyhedral cones, $(a^{l},-\gamma^{l})\in IR^{n}\times IR$. To solve this problem in [9] a modified PCF algorithm is suggested. It finds the a^{l} point in a more efficient way when the set *A* under consideration is not too large. In the modified PCF algorithm at each iteration *l*, the problem (P_l) is solved for each $a_{i}^{l}\in A^{l}$ 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 $l_{0}=\max\left\{l_{i}:i\in I_{l}\right\}$ [9].

C. A BINARY CLASSIFICATION ALGORITHM

The modified PCF algorithm defined in the previous section is more efficient than the old one but it has a hole when one of the training set *A* under consideration is too large. It causes more iterations, at each iteration *l, Pl* is solved for each $a^{i}\in A$.

We solve this problem by using clustering methods. These useful methods for analysis of patterns in data, are offered by data mining, in particular machine learning algorithms. Cluster analysis algorithms form groups of objects that share common properties[11]. Several algorithms have been studied for clustering method [1]. In this paper we use one of respected, *k*-means algorithm.

Given a set of observations $(x_{1},...,x_{m})$ where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition the *m* observations into *k* sets (k ≤ m) $S=\left\{S_{1},...,S_{k}\right\}$ so as to minimize the within-cluster sum of squares (WCSS) [4]:

$$\arg\min_{S} \sum \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2$$

where μ_i is the mean of points in S_i.

The *k*-means algorithm proceeds as follows [4]:

Algorithm 2. *k*-means algorithm.

Step 1. Choose a seed solution consisting of *k* centers (not necessarily belonging to *A*);
Step 2. Allocate data points $a^i \in A$ to its closest center 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 cluster.

In modified PCF algorithm, mentioned in the previous section, the a^l center points are chosen according to the number of elements A^l separated from *B*. In constraint (3), obtaining maximal number of elements is related to the closeness of A^l points to a^l. Therefore instead of solving P_l for each $a_i^l \in A^l$ to find an optimal center in every iteration in modified PCF, we initially use *k*-means clustering method and obtain k numbers of a^k optimal centers that are the closest ones to the corresponding A^k points with reference to the *k*-means method. Then solve k numbers of P_l subproblems for each a^k in PCF algorithm.

Besides, in this paper we change the constraint (4) of P_l subproblem for decreasing large differences between accuricies on training and test sets, and we aim to prevent over-fitting the classification problem and get a good generalization. We apply relaxation to this constraint by allowing some misclassification as follows;

$$-w(b^j - a_k) - \xi \left\| b^j - a_k \right\|_1 + \gamma - 1 \le z_j, \quad j \in J$$

where $z_j > 0$ is a slack variable that measures how much a *B* point fails to be outside of the polyhedron corresponding to the sublevel set $\{x : g_l(x) > 1\}$. If $z_j = 0$, there is no misclassification and if $z_j = 1$, b^j point is on the polyhedron.

Thus, we construct P_l subproblem as follows;

$$\min \frac{1}{m} \sum_{i=1}^{m} y_i + C \frac{1}{p} \sum_{i=1}^{p} z_j$$

$$w(a^i - a_k) + \xi \left\| a^i - a_k \right\|_1 - \gamma + 1 \le y_i, \quad i \in I$$
$$-w(b^j - a^l) - \xi \left\| b^j - a^l \right\|_1 + \gamma - 1 \le z_j, \quad j \in J$$
$$y_i, z_j \ge 0, C \ge 1, w \in R^n, \xi \in R, \gamma \ge 1$$

where $C \ge 1$ is the fixed penalty parameter, used for the misclassification of $b^j \in B$ points, that is predefined.

Then, a binary classification algorithm based on clustering and PCF functions defined as follows:

Let *A* and *B* two given sets in IR^n

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

where $I = \{1,..,m\}$, $J = \{1,..,p\}$.

Algorithm 3. PCF algorithm with clustering for binary classification

Step 0.(Initialization step) Apply clustering algorithm on set of *A*. Let *s* be the number of clusters and *k=1* . I_k=*I*.

Step 1. Let a_k be the center of *k* th cluster . Solve subproblem P_k

$$(P_k) \quad \min \frac{1}{m}\sum_{i=1}^{m} y_i + C\frac{1}{p}\sum_{i=1}^{p} z_j$$

$$w(a^i - a_k) + \xi \left\|a^i - a_k\right\|_1 - \gamma + 1 \le y_i, \quad i \in I_k$$
$$-w(b^j - a_k) - \xi \left\|b^j - a_k\right\|_1 + \gamma - 1 \le z_j, \quad j \in J$$

$$y_i, z_j \ge 0, C \ge 1, w \in R^n, \xi \in R, \gamma \ge 1$$

Let $w_k, \xi_k, \gamma_k, y_k$ be a solution of (P_k). Let

$$g_k(x) = g_{(w_k, \xi_k, \gamma_k, a_k)}(x)$$

Step 2. If $k < s$, let *k=k+1* , $I_k = \left\{i \in I_{k-1} : g_{k-1}\left(a^i\right) > 0\right\}$

and go to *Step 1*.

Step 3. Define the function *g(x)* (separating the sets *A* and *B*) as

$$g(x) = \min_k g_k(x)$$

and stop.

III. RESULTS & DISCUSSION

We present the efficiency of the presented algorithm by carrying out numerical experiments with a number of real world and synthetic datasets. MATLAB is used for applications. We compare proposed algorithm and PCF algorithm due to their accurices and time. The results are shown in tables.

Accuracy is defined as the ratio between the number of well classified points of both A and B as follows:

wc: number of well classified points of A and B

te: number of training set elements

$$\text{Accuracy}=\frac{100\times wc}{te}$$

In the proposed algorithm, *k*-means method was used for clustering. k*(1-20)* was defined as to get the best accuracy. C penalty number was defined as 10 to allow less misclassification to $b^j \in B$ points than $a^i \in A$ points.

The accuricies are not 100 as in modified PCF [9] because of allowing misclassification to $b^j \in B$ points, and stopping the algorithm at the *k* th (number of clusters) iteration.

Table 1 shows the number of instances and attributes of the datasets used. The results shown in Table 2 indicates that the proposed algorithm is more efficient with regard to time. Accuracy values are not 100% as PCF because of allowing some misclassification and stopping the algorithm in *k* (defined in clustering method) iterations. We terminate the algorithm if time exceeds 1800 sec. and show it with (-).

Besides on the same datasets for testing the validities of new algorithm and PCF algorithm, ten-fold cross validation tests are applied. Ten-fold cross validation is explained as followsin [10]; the dataset D is randomly split into 10 mutually exclusive subsets (the folds) $D_1,D_2,..,D_{10}$ of approximately equal size. The inducer is trained and tested 10 times; each time $t \in \{1,2,...,10\}$, it is trained on $D\backslash D_t$ and tested on D_t. The cross validation estimate of accuracy is the overall number of correct classifications, divided by the number of instances in the data set [10].

Table 1. *The brief description of real world data sets*

Data sets	Number of instances	Number of attributes
Blood Transfusion	748	5
Ionosphere	351	34
Fertility	100	10
WBCD	683	9
Heart	297	13
Connectionist Bench	208	60

Table 2. *Results of real-world data sets obtained using Algorithm 3 and PCF*

Data sets	ALGORITHM 3		PCF	
	Accuracy %	Time Sec.	Accuracy %	Time Sec.
Blood Transfusion	60.45	338 sec.	-	-
Ionosphere	98.86	42 sec.	100	756 sec.
Fertility	90	22 sec.	100	35 sec.
WBCD	97.28	35 sec.	100	1763 sec.
Heart	85.90	22 sec.	100	613 sec.
Connectionist Bench	83.80	84 sec.	100	287 sec.

Table 3. *Tenfold cross-validation results of real-world data sets obtained using Algorithm 3*

Data sets	ALGORITHM 3		PCF	
	Training Accuracy %	Testing Accuracy %	Training Accuracy %	Testing Accuracy %
Blood Transfusion	80.56	78.55	-	-
Ionosphere	98.10	94.28	100	95.76
Fertility	90.22	80.23	100	90.45
WBCD	98.21	98.55	100	100
Heart	88.88	91.66	100	86.95
Connectionist Bench	100	96.79	100	80.38

Table 4. *The brief description of synthetic data sets*

Data sets	Number of instances	Number of attributes
Dataset 1	20	6
Dataset 2	50	6
Dataset 3	100	6
Dataset 4	300	6

Table 5. *Accuracy results of synthetic datasets*

Data sets	Modified PCF [17]		Algorithm 3	
	accuracy	Time(sec)	accuracy	Time(sec)
Dataset 1	100	0.38	100	0.16
Dataset 2	100	1.70	100	0.11
Dataset 3	100	6.72	100	2.76
Dataset 4	100	211.09	100	0.92

***Table 6**. Tenfold cross-validation results of synthetic data sets obtained using algorithm 3*

Data sets	Modified PCF in[17]		Algorithm 3	
	Training Accuracy %	Testing Accuracy %	Training Accuracy %	Testing Accuracy %
Dataset 1	100	90	100	95
Dataset 2	100	96	100	100
Dataset 3	100	100	100	100
Dataset 4	100	100	100	100

As can be seen from Table 3 the large differences between accuracies on training and test sets can be reduced by letting some amount (defined by the *C* fixed penalty parameter) of misclassification to *B* points.

Also in Table 5 and Table 6 same comparisons are applied to 4 synthetic datasets, whose number of attributes and instances are shown in Table 4.

IV. CONCLUSION

In this paper, an algorithm is developed for solving the binary classification problems. Algorithm is based on clustering and PCF functions for separating the given finite point sets in *n*-dimensional space. The proposed algorithm retrieves overfitting the classification problem by letting misclassifications and allows saving time while solving problems with large datasets by the help of clustering method.

V. REFERENCES

[1] M.R. Anderberg, *Cluster Analysis for Applications,* Academic Press, New York (1973).

[2] A. Astorino, M. Gaudioso, *Journal of Optimization Theory and Applications* **112(2)** (2002) 265.

[3] A.M. Bagirov, *Optimization Methods and Software* **20(2-3)** (2005) 277.

[4] A.M. Bagirov, K. Mardaneh, *WISB '06 Proceedings of the 2006 workshop on Intelligent systems for bioinformatics* **(73)** (2006) 23.

[5] A.M. Bagirov, J. Ugon (2005) **DOI: 10.1007/0-387-26771-9_6**.

[6] A.M. Bagirov, J. Ugon, D. Webb, B. Karasözen, *Pattern Analysis and Applications* **14(2)** (2011) 165.

[7] A.M. Bagirov, J. Ugon, D. Webb, G. Öztürk, R. Kasımbeyli (2011) **DOI: 10.1007/s11750-011-0241-5.**

[8] K.P. Bennett, O.L. Mangasarian, *Optimization methods and software* **1(1)** (1992) 23.

[9] R.N. Gasimov, G. Öztürk *Optimization Methods and Software* **21(4)** (2006) 527.

[10] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, **International Joint Conference on Artificial Intelligence,** San Francisco (1995) 1137.

[11] A. Kusiak (2001) **DOI:10.1117/12.417237**.

[12] J.M. Liittschwager, C. Wang, *Management Science* **24(14)** (1978) 1515.
[13] O.L. Mangasarian, *Operations Research* **13(3)** (1965) 444.
[14] J.B. Rosen, *Journal of Mathematical Analysis and Applications* **10(1)** (1965) 123.
[15] V. Vapnik, *The nature of statistical learning theory,* Springer Verlang, New York (1995).