Digital Signal Processing ••• (••••) •••–•••

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Incremental conic functions algorithm for large scale classification problems

Emre Cimen [a,b,*], Gurkan Ozturk [a,b], Omer Nezih Gerek [a]

[a] *Anadolu University, Faculty of Engineering, 26555, Eskisehir, Turkey*
[b] *Anadolu University, Computational Intelligence and Optimization Laboratory (CIOL), 26555, Eskisehir, Turkey*

A R T I C L E   I N F O

Article history:
Available online xxxx

Keywords:
Polyhedral conic functions
Mathematical programming
Classification
Machine learning

A B S T R A C T

In order to cope with classification problems involving large datasets, we propose a new mathematical programming algorithm by extending the clustering based polyhedral conic functions approach. Despite the high classification efficiency of polyhedral conic functions, the realization previously required a nested implementation of k-means and conic function generation, which has a computational load related to the number of data points. In the proposed algorithm, an efficient data reduction method is employed to the k-means phase prior to the conic function generation step. The new method not only improves the computational efficiency of the successful conic function classifier, but also helps avoiding model over-fitting by giving fewer (but more representative) conic functions.

1. Introduction

The ever-increasing internet bandwidth together with mass storage has long required mining, clustering and classification [1], for applications ranging from finger print or iris recognition based security technologies to detecting spam e-mails, to gesture or face recognition. These technologies improve innovative aspects of commercial products or improve the convenience of the provided service. Consequently, researchers have been interested in classification problems for years. The early research on classification focused mostly on feature extraction and classifier optimization by means of finding best separating functions/surfaces. Unfortunately, the ever-increasing data amount enforced researchers to adopt different strategies to handle the new problem of big data. According to a not-so-new IDC Digital Universe Study [2] collected data doubles in every two years. Nowadays, it is argued that the doubling period is also shrinking. Because of this, developed algorithms must be appropriate to work with large datasets for both in training step (classifier construction) and test step (application). Any improvement to reduce computation times by multiple-folds would be welcome.

Naturally, a necessary property of classification algorithms is its classification accuracy. In that aspect, the developed algorithms are expected to perform well with respect to standard classifiers such as Bayesian classifiers [3], artificial neural networks [4], decision trees [5,6], and support vector machines [7].

Starting from 1960's there has been an interest to classification algorithms based on mathematical programming. A literature example is by Bennett and Mangasarian, where a robust approach for linear separation was developed [8]. In another work, Astorino and Gaudioso used more than one hyper planes to separate two sets; which were found with mathematical programming [9]. Max–Min separation is another related successful approach developed by Bagirov [10]. Similarly, Uney and Turkay developed an Integer Programming algorithm to classify more than one classes with hyper boxes [11]. One of the most famous and commonly used classification algorithms is Support Vector Machines (SVMs), which is based on quadratic programming methods [7]. A survey based on SVMs and its latest improvements can be found in [12]. In [13] non-smoothness in classification, in [14] non-linear programming in classification, in [15] margin maximization based on polyhedral separability, and in [16] ellipsoidal separation for classification problems were investigated. Finally, classification with truncated l_1 distance kernel was introduced in [17]. This presented work is also considered within the class of classification algorithms that use LP.

A critical idea that is utilized in this paper is to construct and use Polyhedral Conic Functions (PCFs), which were first proposed for classification by Gasimov and Ozturk [18] and the classification method was also named PCF algorithm [1]. The authors then combined the PCF approach with Bagirov's Max–Min separation algorithm [19]. Later, clustering based PCF algorithm [20] was developed and successfully applied to real life problems such as arrhythmia classification [21] and gesture recognition [22]. An incremental

* Corresponding author.
E-mail address: ecimen@anadolu.edu.tr (E. Cimen).

https://doi.org/10.1016/j.dsp.2017.11.010

Please cite this article in press as: E. Cimen et al., Incremental conic functions algorithm for large scale classification problems, Digit. Signal Process. (2018), https://doi.org/10.1016/j.dsp.2017.11.010

piecewise linear classification algorithm based on polyhedral conic separation was also introduced in [23].

In this work, the highly accurate clustering based PCF [20] is considered as a starting point, and the part of the creation of the LP problem, to construct the classifier, is smartly modified to improve the computational efficiency, which is expected to enable processing large datasets. The clustering based PCF is a novel method which outperforms several state-of-the-art classifiers, including SVM [20]. However, in its original version, a linear constraint was existing in the LP model for each additional data point. Consequently, when data size is very large, the model had too many constraints to be handled. In this work, we eliminate unnecessary data points (hence constraints) after the clustering stage. The new strategy is observed to keep the high accuracy of PCFs on datasets from UCI Machine Learning Repository while reducing computation times and avoiding over-fitting with fewer conic functions.

Before we introduce the ICF algorithm, we will clarify the notation. The 2-d matrices, which are the sets in this paper, like **A** are symbolized by capital boldface letters. Vectors are represented by boldface lower case letters and scalars are represented by italic lower case letters.

The rest of the paper is organized as follows. In Section 2, the proposed algorithm is explained with algorithmic layouts and example illustrations. In Section 3, experimental results are given. Finally, conclusions are provided. The implemented software of the work is also provided in the accompanying SoftwareX part of this Special Issue [24].

2. Incremental conic functions (ICF) algorithm

2.1. Preliminaries

In this subsection, we briefly describe the notions of PCFs and polyhedral conic separation. More detailed description can be found in [18].

Let **A** and **B** be given disjoint sets in R^n containing m and p points, respectively:

$$\mathbf{A} = \{\mathbf{a}^1, \ldots, \mathbf{a}^m\}, \quad \mathbf{a}^i \in R^n,\ i = 1, \ldots, m, \tag{1}$$

$$\mathbf{B} = \{\mathbf{b}^1, \ldots, \mathbf{b}^p\}, \quad \mathbf{b}^j \in R^n,\ j = 1, \ldots, p. \tag{2}$$

PCFs construct a separation function for the sets **A** and **B** as following.

Definition *([18])*. A function $g : R^n \to R$ is called *conic function* if its graph is a cone and all its level sets satisfy:

$$S(\alpha) = \{\mathbf{x} \in R^n : g(\mathbf{x}) \le \alpha\}, \tag{3}$$

for $\alpha \in R$, yielding, so called, convex sets.

Given $\mathbf{w}, \mathbf{c} \in R^n$, $\xi, \gamma \in R$, the general form of conic function $g_{(\mathbf{w},\xi,\gamma,\mathbf{c})}$: $R^n \to R$ is defined as follows:

$$g_{(\mathbf{w},\xi,\gamma,\mathbf{c})}(\mathbf{x}) = \mathbf{w}(\mathbf{x} - \mathbf{c}) + \xi \|\mathbf{x} - \mathbf{c}\|_p - \gamma, \tag{4}$$

where $\|\mathbf{x}\|_p$ is an l_p-norm of the vector $\mathbf{x} \in R^n$ and $g(\mathbf{x})$ defines a conic function that can be used for constructing discriminating regions of two arbitrary sets: **A** and **B**. Here, it must be noted that **x** is a point in the set and **c** is the Euclidian center (or selected from R^n in some approaches) (i.e., not the l_p center) of the set, that was already calculated before the solution of the LP. This center also corresponds to the lowest (vertex) point of the cone. The degree (p) of l_p norm can be varied to obtain a rich class of convex sets, ranging from 1 to ∞. In Fig. 1, level sets (horizontal intersection with the plane $z = 0$) of various cones are illustrated for $p = 1$, 2, 4, 10 and 50 and $\mathbf{w}^T = [1, 1]$, $\mathbf{c}^T = [0, 0]$, $\gamma = 2$.

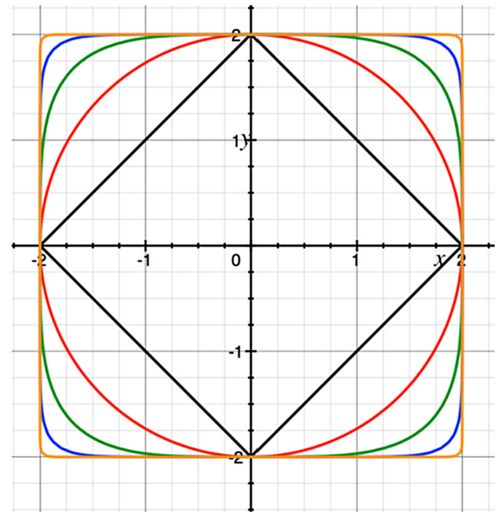

Fig. 1. Level sets of conic function for $p = 1, 2, 4, 10$ and 50, and $\mathbf{w}^T = [1, 1]$, $\mathbf{c}^T = [0, 0]$, $\gamma = 2$.

Throughout this work, l_1-norm is used to define a particular conic shape, which is the "polyhedral conic shape":

$$g_{(\mathbf{w},\xi,\gamma,\mathbf{c})}(\mathbf{x}) = \mathbf{w}(\mathbf{x} - \mathbf{c}) + \xi \|\mathbf{x} - \mathbf{c}\|_1 - \gamma. \tag{5}$$

Using multiple different polyhedral conic functions, it becomes possible to separate and classify arbitrarily distributed (non-convex) sets **A** and **B**. PCF algorithms [18], k-means based PCF algorithm [20] and incremental PCF algorithm [23] have different center selection and updating strategies. In [18], the center points are randomly selected from the set **A** and classification was achieved by sequentially eliminating correctly classified points. In [20], the centers of PCFs are found by solving k-means algorithm and then a PCF is obtained by solving LP for each cluster. In [18] and [20] an eventual classifier is constructed as point-wise minimum of all PCFs, requiring several PCF constructions. In [23], classifier is obtained by minimizing a single error function in an incremental manner.

2.2. Review of k-means based polyhedral conic functions (PCFs)

In the k-means based Polyhedral Conic Function approach [20], the classifiers are constructed with the simultaneous use of the polyhedral conic separation approach [18] and the k-means clustering technique. This algorithm applies k-means algorithm to find centers of PCFs. In order to construct the classifier for a specific class, the class is first divided into sub-clusters via k-means algorithm. Then, an LP is solved for each cluster in order to obtain a PCF that separates the cluster from the other classes. Thus, k PCFs are obtained after this operation. The classifier of the selected class is obtained as a point-wise minimum of k separate PCFs. These steps are repeated for each class. If the number of classes is η, the total number of constructed PCFs is $\eta \times k$. In the test phase, a data point is applied to each of these $\eta \times k$ PCFs, and the class η_i is chosen for the PCF function which yields the minimum function value. In the original form of the polyhedral conic separation method [18], the number of PCFs to be tested was huge as compared to the limited ($\eta \times k$) number of PCFs. Therefore, the use of clustering algorithm allows to significantly decrease the number of centers and consequently the number of PCFs which makes the algorithm reasonable for real life applications and helps to avoid over-fitting problem.

Please cite this article in press as: E. Cimen et al., Incremental conic functions algorithm for large scale classification problems, Digit. Signal Process. (2018), https://doi.org/10.1016/j.dsp.2017.11.010