Hindawi
Scientific Programming
Volume 2018, Article ID 5349284, 11 pages
https://doi.org/10.1155/2018/5349284

Research Article

Application of the Polyhedral Conic Functions Method in the Text Classification and Comparative Analysis

Nur Uylaş Satı [1] **and Burak Ordin** [2]

[1] *Bodrum Maritime Vocational School, Muğla Sıtkı Koçman University, Bodrum, Muğla, Turkey*
[2] *Department of Mathematics, Faculty of Science, Ege University, Izmir, Turkey*

Correspondence should be addressed to Nur Uylaş Satı; nuruylas@gmail.com

Received 15 January 2018; Revised 17 April 2018; Accepted 20 May 2018; Published 28 June 2018

Academic Editor: José M. Lanza-Gutiérrez

In direct proportion to the heavy increase of online information data, the attention to text categorization (classification) has also increased. In text categorization problem, namely, text classification, the goal is to classify the documents into predefined classes (categories or labels). Recently various methods in data mining have been experienced for text classification in literature except polyhedral conic function (PCF) methods. In this paper, PCFs are used to classify the documents. The separation algorithms via PCFs which include linear programming subproblems with inequality constraints are presented. Numerical experiments are done on real-world text datasets. Comparisons are made between state-of-the-art methods by presenting obtained tenfold cross-validation results, accuracy values, and running times in tables. The results verify that in text classification PCF methods are as effective in terms of accuracy values as state-of-the-art methods.

1. Introduction

The supervised data classification is one of the essential fields in data mining. The researches regarding this field deal with the categorization of data for its most effective and efficient use. The objective of supervised data classification is to determine rules on the training set for the data classification. This set consists of some features of data whose labels (classes or categories) are known. To discover the system, training subsets of the given dataset are used and utility of the obtained rules is examined on the test set. It has so many application areas such as medicine, engineering, business, and education [1–4]. Various learning algorithms for supervised data classification have been defined in machine learning. For instance, linear regression, logistic regression, decision tree, support vector machines, Naive Bayes, K-nearest neighbour, K-means, random forest, dimensionality reduction algorithms, and gradient boost and adaboost are the most commonly used ones [5].

The process of supervised data classification, where the dataset consists of text data, is called text classification. With the heavy increase of online information, it has been so difficult to control, present, and archive the text data uniformly. Text classification has been one of the main techniques for organizing text data and it is used for classifying columns and news in terms of their subjects, to help a user's search on hypertext, to surf on the Internet, and so forth. Because finding text classifiers by hand is gruelling and time-consuming, data mining techniques are utilized in text classification [6, 7].

For text classification, besides the commonly used supervised classification techniques, we wish to experience polyhedral conic functions as supervised classification functions $f : X \to C$ that map documents to labels (classes) [8]. In the following state-of-the-art review, we sketch out some of learning techniques used for text categorization in literature. The process of text classification will be examined and mathematical model of a text classification problem will be presented in Section 3. In the fourth section, polyhedral conic functions are explained and utilization of these functions in data classification will be mentioned by presenting the algorithms in literature. In the fifth section, defined

algorithms via polyhedral conic functions are regulated for text categorization problems. In the sixth section, numerical experiments are done by implementing defined algorithms on a determined real-world dataset. Obtained running time, training, and test accuracy values are presented in tables. Also for comparison with state-of-the-art methods and to see the efficiency of defined algorithms on large datasets, implementations are made on various real-world datasets from UCI (machine learning repository). Finally in the last section the paper is concluded.

2. Related Works

In the literature, several authors have proposed approaches for text classification problem. Text categorization (text classification) is the process of automatically labeling a set of documents into classes (categories) by using predefined training dataset. The researchers are so interested in text classification studies because of the development of technology and increase in the number of the electronic documents available in several sources. The whole process of text classification has some steps that will be introduced in the third section. In our study, we focus on the step of data mining (learning models). Since we work on a supervised learning model in text classification, in this section of related works, we sketch out some of machine learning techniques commonly used in literature in training a text classification model by explaining the approaches that they use.

K-nearest classifier method is based on the hypothesis of the class (category or label) of a sample that is most similar to the class of other samples that are closest in the vector space. The training sets are viewed in multidimensional feature space. Here, the training set is divided into zones in terms of the defined classes. In the feature space, an instance is assigned to a specific class if it is the most proper class among the number of k-nearest training data. Commonly Euclidean Distance is used as distance metric between the points. This method is usable since various similarity measures can be used for describing neighbours of an instance [9]. A comparative study of KNN and SVM methods was done in [10]. And also in [11–13], KNN method in text classification is examined.

Rocchio's method is a vector space method for document filtering or routing in informational retrieval. In this method, a prototype vector for each class is created by the help of training set, for instance, the mean vector of points in class of c_i. Similarity between test data (document) and each of prototype vectors is calculated. Finally test data is assigned to the class with maximum similarity [14]. In [15–17], this method is examined for text categorization and information retrieval. In [18], a new algorithm called HI-Rocchio is proposed. This algorithm combines two methods: Rocchio's method and Hierarchical clustering. In their experimental results, they verified the effectiveness of the algorithm.

Naive Bayes method is based on probability. The optimal class in NB method is the most likely or maximum a posteriori (MAP) class c_{map}:

$$C_{map} = \arg\max P(c \mid d) = \arg\max P(c) \prod_{1 \le k \le n_d} P(t_k \mid c). \quad (1)$$

Here d is adocument; $c \in C$ is a predicted class where $C = c_1, c_2, \ldots, c_J$ is a fixed set of classes. $P(t_k \mid c)$ is a measure of how much evidence t_k contributes that c is the right class. $P(c)$ is the prior probability of a document that belongs to class of c [9].

In [19–22], NB method is examined and performance of NB algorithms is compared with other learning methods.

The decision tree method uses the form of a tree structure for classification of training documents. In the structure of a decision, leaves symbolize the class of documents and branches symbolize connectors of features that conduct to those categories [10]. In [10, 23, 24], decision tree models in text categorizations are examined.

Support vector machine (SVM) is a machine learning method defined by V. Vapnik et al. in 1990. Discriminant-based optimization is used and linear separator parameters are found by using labeled datasets in this method. SVM method is utilized by many researchers in different areas [25]. In [6, 7, 10, 12] SVM learning method is studied for text categorization and comparisons with other learning methods in different datasets are proposed. In [26], news articles are used to predict intraday price movements of financial assets by using SVMs algorithm in training process with a given kernel matrix. Multiple kernel learning is used to combine equity returns with text as predictive features. It is seen that text features producing significantly better performance than historical returns alone.

Classification via regression method uses regression methods for classification. Class is binarized and one regression model is built for each class value. In [22] classification via regression is used for detection of child exploiting chats from a mixed chat dataset as a text classification task and it is seen that Naive Bayes and this method compete each other such that they detect almost the same number of child exploitation chats.

In addition to these, text classification is studied by combining text classifiers by different researchers to improve the efficiency of classification. In [27], Fragos K. et al. combined the methods that belong to the same paradigm-probabilistic. Naive Bayes and maximum entropy classifiers are combined to test on the applications where the individual performance is good. In [28], S. Keretna et al. combined the individual results of Conditional Random Field (CRF) classifiers and maximum entropy (ME) classifiers on the medical text. They all get better performance results than the individual classifiers. All the combined text classifiers till 2016 are reviewed in [29].

In [30], all these methods are compared and discussed with their improvements. The authors see that each researcher has their own datasets for testing the improvement which makes the comparison more difficult. Because of this reason, in this paper, besides our own dataset for testing, commonly used and easily accessible benchmark datasets are used in the testing phases.

The most recent article that overviews the state-of-the-art elements in text classification is published by Mironczuk M. and Protasiewicz J. in [31]. They reviewed the works dealing with text classification according to data collection, data analysis for labeling, feature construction and weighting, feature selection and projection, training of a classification model, and solution evaluation. They found numerous papers on the issue of training algorithms in text classification [32–35]. In their work, they found two more training methods of a classification function in the literature different from the above given approaches: neural network classifier and artificial immune systems studied, respectively, in [33, 36].

In this study, we experiment the data mining process of text classification by using a different classifier as distinct from above approaches in literature. We aim to get better performance results than the previous approaches, by using mathematical programming and utilizing polyhedral conic functions in training algorithm of text categorization process.

3. Text Classification

The solution of data classification problem consists of two steps. In the first step, a classifier function which describes a predetermined set of data classes is built. It is called learning step on training set. A classification algorithm builds the classifier by analyzing a training set made up of a dataset and its associated class labels. In the second step, obtained classifier function is tested on a test set. The effectiveness of a classifier function is determined by the evaluation process. All these steps and preparation processes are explained in the following paragraphs for text classification task.

Text classification, namely, text categorization, aims at classifying the documents into a fixed number of predefined classes (labels). In order to get good text classification results, the choice of a proper and effective algorithm plays an important role. Merely, the whole process of text classification should not be ignored. The steps of this process can be given as follows:

(i) Determining of text data collection
(ii) Text preprocessing
(iii) Attribute selection
(iv) Text transformation
(v) Data mining
(vi) Evaluation

In determining of text data collection, document datasets (like html, pdf, doc, web content, etc.) are constituted. These datasets consist of many words.

In text preprocessing, the text documents are presented into clear word format, e.g., expression to express, behaviour to behave. These words are cleaned out from stop words, conjunctions, and meaningless expressions, and then roots of words are determined. Commonly the steps taken in text preprocessing are Tokenization and Removing Stop Words like frequently occurring "the", "and", etc. [37].

In attribute selection part, important words in preprocessed documents are detected and nonrelevant words, for instance, words that are placed in the whole documents or nearly in all of documents, are eliminated.

In text transformation, documents are defined with a goal-oriented suitable representation for learning algorithm. Namely, unstructured data should be transformed into structured data. Here the aim is to reduce the complexity of the documents for an easy managing procedure by transforming the full text version of the document to a document vector. Vector space model (SMART) where documents are represented by vectors of words is the commonly used document representation. Some of the limitations of this model are high dimensionality of the representation, loss of correlation with adjacent words, and loss of semantic relationship that exists among the terms in a document. To overcome these problems, term weighting methods are used to assign appropriate weights to the term [37].

In vectorial representation, the term-document, $d \times t$, matrix is created; here d represents the numbers of documents and t represents the numbers of the terms. The value in the (i,j)th entry of $d \times t$ matrix stands for the density of jth term in ith document. By using $d \times t$ matrix, any documents from the collection can be represented by various methods such as bag of words, vector space model (SMART).

The used document in this paper is represented by vectorial using. $TF(i,j)$, that is called term density, is the weight of jth term in ith document. $IDF(j)$, that is called inverse document density, is the weight of jth term in all collection for a $d \times t$ term-document matrix. Classical formula of TF-IDF is as follows:

$$w(i,j) = TF(i,j) \times IDF(j), \tag{2}$$

where $0 \leq w(i,j) \leq 1$, $0 \leq TF(i,j) \leq 1$, $0 \leq IDF(i,j) \leq 1$. Here $w(i,j)$ is called the weight of jth term in ith document.

In data mining step, a proper and effective method and algorithm are chosen and implemented to the transformed dataset. Some methods as Naive Bayes, Rocchio's method, and k-nearest classifier are used for data classification of text data. Besides we foresee that the separation via PCFs methods based on mathematical optimization can be applicable on text data. So we experiment the PCFs separation algorithms on a real-world dataset in this paper. Separation with PCFs is expressed in detail in Section 4.

Mathematical model of a binary classification problem can be introduced as linear separability or polyhedral separability. They are explained as follows in [38].

Let A and B be given sets containing $m \in Z^+$ and $p \in Z^+$ n-dimensional vectors, respectively:

$$\begin{aligned} A &= \left\{a^1, \ldots, a^m\right\}, \\ a^i &\in R^n, \quad i = 1, \ldots m, \\ B &= \left\{b^1, \ldots, b^p\right\}, \\ b^j &\in R^n, \quad j = 1, \ldots p. \end{aligned} \tag{3}$$

The sets A and B are **linearly separable** if there is a hyperplane $\{x, y\}$, with $x \in R^n$, $y \in R^1$ such that,

for any $i=1,\ldots, m$,

$$\left\langle x, a^i \right\rangle - y \le 0, \tag{4}$$

for any $j=1,\ldots, p$,

$$\left\langle x, b^j \right\rangle - y > 0. \tag{5}$$

A characterization of linear separability is that the convex hulls of the two sets do not intersect. If the intersection is not empty, it is possible to obtain a hyperplane that minimizes some misclassification measure or even to look for nonlinear separating surfaces. The problem of finding this hyperplane is formulated as the following optimization problem [39]:

$$\begin{aligned} &\min \quad f(x, y) \\ &\text{subject to} \quad (x, y) \in R^{n+1}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} f(x, y) = {} & \frac{1}{m}\sum_{i=1}^{m} \max\left(0, \left\langle x, a^i \right\rangle - y + 1\right) \\ & + \frac{1}{p}\sum_{j=1}^{p} \max\left(0, -\left\langle x, b^j \right\rangle + y + 1\right) \end{aligned} \tag{7}$$

is an error function. Here $< .,. >$ stands for the scalar product in R^n. It is shown that the given minimization problem is equivalent to the following linear program [39]:

$$\frac{1}{m}\sum_{i=1}^{m} t_i + \frac{1}{p}\sum_{j=1}^{p} z_j, \tag{8}$$

subject to

$$\begin{aligned} \left\langle x, a^i \right\rangle - y + 1 < t_i, \quad & i = 1, \ldots, m, \\ -\left\langle x, b^j \right\rangle + y + 1 < z_j, \quad & j = 1, \ldots, p, \\ t \ge 0, \ z \ge 0, & \end{aligned} \tag{9}$$

where t_i is nonnegative and shows the error for the data $a^i \in A$ and z_j is nonnegative and shows the error for the data $b^j \in B$.

The concept of ***h* polyhedral separability** was introduced in [40]. The sets A and B are h polyhedrally separable if there is a set of h hyperplanes $\{x^i, y_i\}$, with

$$\begin{aligned} & x^i \in IR^n, \\ & y_i \in IR^1, \\ & i = 1, \ldots, h, \ h \in Z^+, \end{aligned} \tag{10}$$

such that

(1) for any $j = 1, \ldots, m$ and $i = 1, \ldots, h$

$$\left\langle x^i, a^j \right\rangle - y_i < 0, \tag{11}$$

(2) for any $k = 1, \ldots, p$ there is at least one $i \in \{1, \ldots, h\}$ such that

$$\left\langle x^i, b^k \right\rangle - y_i > 0. \tag{12}$$

The problem of polyhedral separability of the sets A and B is reduced to the following problem [40]:

$$\begin{aligned} &\text{minimize} \quad f(x, y) \\ &\text{subject to} \quad (x, y) \in IR^{(n+1)\times h}, \end{aligned} \tag{13}$$

where

$$\begin{aligned} & f(x, y) \\ & = \frac{1}{m}\sum_{j=1}^{m} \max\left[0, \max_{1\le i\le h}\left\{\left\langle x^i, a^j \right\rangle - y_i + 1\right\}\right] \\ & \quad + \frac{1}{p}\sum_{k=1}^{p} \max\left[0, \min_{1\le i\le h}\left\{-\left\langle x^i, b^k \right\rangle + y_i + 1\right\}\right] \end{aligned} \tag{14}$$

is an error function. In [40], also an algorithm for solving defined minimization problem is developed. The calculation of the descent direction at each iteration of this algorithm is reduced to a certain linear programming problem.

Besides, all introduced mathematical optimization techniques can be applied for multiclass classification problems, where we have more than two classes, by using one versus all strategy. This means that for given dataset A with $q \ge 2$ classes $A_1, \ldots, A_q$, any class A_j, $j \in \{1, \ldots, q\}$, is taken as the set A and the set B is defined as a union of all remaining classes [41].

In a text classification problem, a definition $d \in X$ of a document is given; here X is the document space that includes blog posts, news stories, articles, web pages, and technical reports; and a constant set of classes $C = \{C_1, C_2, \ldots, C_m\}$. The classes are in general subjects, authors, and topics but may also be based on types and interests. Classes are human defined for needs of the problem. This is a supervised learning problem since we study with a given training set T of labeled document shown in

$$T = \{(d_1, c_1), \ldots, (d_m, c_m)\} \quad d \in X, \ c \in C. \tag{15}$$

For example, *(d,c) =(mathematical optimization, life sciences)* indicates that mathematical optimization document is labeled with life sciences.

When we turn back to the subject of representation of the document collection, since we are working on supervised classification, we should add a new column to d×t matrix such that the value in last column represents the classes of the documents. Thus we use a *d×(t+1)* matrix during the text classification algorithm. Here d is the number of documents and t is the number of attributes (e.g., word stems).

Here, the objective is to find rules (functions) under favour of training set, *d×(t+1)* matrix, and evaluate the efficiency of the obtained rules (functions) on the test set.

Correspondingly the text classification problem's dimension is directly related to the number of documents and

the word stems exist in the whole document collection that constitutes $d\times(t+1)$ matrix.

In performance evaluations, many measures have been used, such as F-measure, fallout, error, and accuracy. In this paper, accuracy values of training and testing phases are calculated by implying cross-validation method. These subjects will be viewed in detail in Section 6.

In the following section, an approximation via polyhedral conic functions based on mathematical optimization is expressed.

4. Classification via Polyhedral Conic Functions (PCFs)

Polyhedral conic functions (PCFs) have been introduced in 2006 by Gasimov and Öztürk to separate two different labeled point sets, in other words, to split two discrete datasets [8]. Every point is represented with a vector whose every index except the last corresponds to an attribute of a point (data) and the last index stands for the class (label) of the point.

Polyhedral functions are defined as follows in [8]:

$$g_{(w,\xi,\gamma,a)} : IR^n \longrightarrow IR = w'(x-a) + \xi \|x-a\|_1 - \gamma, \tag{16}$$

where x is an n-dimensional point (vector), $x, w, a \in IR^n, \xi, \gamma \in IR, w'x = w_1x_1 + \ldots + w_nx_n, \|x\|_1 = |x_1| + \ldots + |x_n|$.

Definition 2 and Lemma 1 quoted below are given and proved in [8].

Lemma 1. *A graph of the function $g_{(w,\xi,\gamma,a)}$ defined in (16) 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.*

It follows from Lemma 1 that every polyhedral function given in (16) performs as a polyhedral conic function (PCF).

Definition 2. 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.

The first separation algorithm via PCFs was defined in [8] as follows:

Let A and B be given sets containing $m \in Z^+$ and $p \in Z^+$ n-dimensional vectors, respectively:

$$A = \left\{a^i \in R^n, i \in I\right\},$$
$$B = \left\{b^j \in R^n, j \in J\right\} \tag{17}$$
$$\text{where } I = \{1, \ldots, m\}, \; J = \{1, \ldots, p\}.$$

Algorithm 3. Binary classification via PCFs.

Step 0 (initialization step). Let l=1, I_l=I, A_l =A and go to Step 1.

Step 1. Let a_l be an arbitrary point of A. Solve subproblem (P_l).

$$(P_l) \quad \min\left(\frac{y'e_m}{m}\right), \tag{18}$$

$$w'\left(a^i - a^l\right) + \xi\left\|a^i - a^l\right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l, \tag{19}$$

$$-w'\left(b^j - a^l\right) - \xi\left\|b^j - a^l\right\|_1 + \gamma + 1 \le 0, \quad \forall j \in J, \tag{20}$$

$$y = (y_1, \ldots, y_m) \in R_+^m, \; w \in R^n, \; \xi \in R, \; \gamma \ge 1. \tag{21}$$

Let $w^l, \xi^l, \gamma^l, y^l$ be a solution of (P_l). Let

$$g_l(x) = g_{(w^l,\xi^l,\gamma^l,a^l)}(x). \tag{22}$$

Step 2. $I_{l+1} = \{i \in I_l : g_l(a^i) + 1 > 0\}, A_{l+1} = \{a^i \in A_l : i \in I_{l+1}\}, l = l + 1$. If $A_l \neq \emptyset$ go to Step 1.

Step 3. Determine the function $g(x)$ (parting the sets A and B) as

$$g(x) = \min_l g_l(x), \tag{23}$$

and stop.

This algorithm was modified for binary classification problems in [42, 43]. Clustering algorithm is added to the initialization step to decrease running time by reducing the step size that is required for finding the center points of polyhedral conic functions. Clustering algorithms form groups of objects that share common properties [44]. Several algorithms have been studied for clustering method [45, 46]. In [43], one of the most efficient clustering algorithms, k-means method, was used and also in [42], k-medoids method that differs from k-means in the determined center points' features was experienced. Besides, relaxation was applied to (P_k) subproblem constraint (20) to avoid extra variations between accuracy values of training and test sets (called overfitting) by allowing (z_j) misclassification as in (26). In conjunction with the applied change P_l subproblem (18) is changed as in (24). The modified PCF algorithm was defined in [43] as follows.

Algorithm 4. Binary classification via PCFs and clustering method.

Step 0 (initialization step). Apply ***k*-means** clustering algorithm over 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 kth cluster. Solve subproblem P_k.

$$(P_k) \quad \min \frac{1}{m}\sum_{i=1}^{m} y_i + C\frac{1}{p}\sum_{i=1}^{p} z_j, \tag{24}$$

$$w\left(a^i - a_k\right) + \xi\left\|a^i - a_k\right\|_1 - \gamma + 1 \le y_i, \quad i \in I_k, \tag{25}$$

$$-w\left(b^j - a_k\right) - \xi\left\|b^j - a_k\right\|_1 + \gamma - 1 \le z_j, \quad j \in J, \tag{26}$$

$$y_i, z_j \ge 0, \; C \ge 1, \; w \in R^n, \; \xi \in R, \; \gamma \ge 1. \tag{27}$$

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

$$g_k(x) = g_{(w_k,\xi_k,\gamma_k,a_k)}(x). \quad (28)$$

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

Step 3. Determine the function $g(x)$ (parting the sets A and B) as

$$g(x) = \min_k g_k(x), \quad (29)$$

and stop.

5. PCF Algorithms for Text Categorization

In this paper, PCF algorithms are used for text categorization. Algorithms 3 and 4 are both defined for binary classification problems; merely lots of text categorization problems include more than two classes so we should use the multiclass classification algorithms. The only difference between binary and multiclass classification problems is the number of the classes. For this reason binary classification methods can be simply adapted to multiclass classification problems by applying Algorithm 3 or 4 (binary classification algorithm) between each class and the rest. The number of classifiers formed during the algorithm is "*n.k*"; here "*n*" is the number of classes and "*k*" represents the number of clusters. In every iteration, binary classification algorithm is implemented to *Aj, j=1,2,. . .,n* and *A\Aj* sets so "*k*" different $g_j^{1,\ldots,k}$ classifiers are formed. In testing phase, the class of "a" point is defined by

$$j = \arg\min g_j^{1,\ldots,k}(a). \quad (30)$$

Therefore, the finisher separating function is identified as the pointwise minimum of all functions that is formed after binary classifications:

$$g(x) = \min_j g_j^{1,\ldots,k}(x), \quad j = 1, \ldots, n. \quad (31)$$

A multiclass classification algorithm, using clustering method and polyhedral conic functions, is defined as follows in [42].

Algorithm 5. Multiclass classification algorithm using clustering method and PCFs.

Step 0 (initialization). Let $A = A_1 \cup A_2 \cup \ldots \cup A_c$, $A = \{a_l^i \in IR^n : i \in I_l, l = 1, 2, \ldots, c\}$, l=1.

Step 1. $B = A/A_l$, $B = \{b_l^j \in R^n : j \in I/I_l\}$.

Step 2. Apply clustering algorithm in A_l. Let k be the number of clusters and s=1, $I_l^1 = I_l$, and $A_l^1 = A_l$.

Step 3. Let $a_s \in A_l^s$ be the sth center of A_l. Solve P_l^s subproblem.

$$(P_l^s) \quad \min\left(\frac{y' e_{|I_l|}}{|I_l|}\right), \quad (32)$$

$$w'\left(a^i - a_s\right) + \xi\left\|a^i - a_s\right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l^s, \quad (33)$$

$$-w'\left(b_l^j - a_s\right) - \xi\left\|b_l^j - a_s\right\|_1 + \gamma + 1 \le 0, \quad \forall j \in \frac{I}{I_l}, \quad (34)$$

$$y = \left(y_1, \ldots, y_m\right) \in R_+^{I_l^s}, \; w \in R^n, \; \xi \in R, \; \gamma \ge 1. \quad (35)$$

Let $w_s, \xi_s, \gamma_s, y_s$ be the solution of (P_l^s),

$$g_l^s(x) = g_{(w_s,\xi_s,\gamma_s,a_s)}(x). \quad (36)$$

Step 4. If $s < k$, let s=s+l, $A_l^s = \{a^i \in A_l^{s-1} : g_l^s(a^i) > 0\}$, $I_l^s = \{i \in I_l^s : a^i \in A_l^s\}$ and go to Step 3.

Step 5. If l<c, let l=l+*1* and go to Step 1.

Step 6. Determine the function *g(x)* parting A_l, *l=1,. . ., c,* as follows:

$$g(x) = \min_l g_l^{1,\ldots,k}(x), \quad (37)$$

and stop.

Algorithm 5 is constituted from Algorithm 4 but z_j misclassifications are not added as in (26) constraint; it is abandoned as in (20) constraint of Algorithm 3. In [47], the z_j added form of Algorithm 5 is defined as follows.

Algorithm 6. Multiclass classification algorithm that allows misclassifications for both of the sets besides clustering method and PCFs.

Step 0 (initialization). Let $A = A_1 \cup A_2 \cup \ldots \cup A_c$, $A = \{a_l^i \in IR^n : i \in I_l, l = 1, 2, \ldots, c\}$, l=1.

Step 1. $B = A/A_l$, $B = \{b_l^j \in R^n : j \in I/I_l\}$.

Step 2. Apply clustering algorithm in A_l. Let k be the number of clusters and s=1, $I_l^1 = I_l$, and $A_l^1 = A_l$.

Step 3. Let $a_s \in A_l^s$ be the sth center of A_l. Solve P_l^s subproblem.

$$(P_l^s) \quad \min \frac{1}{|A_l^s|}\sum_{i \in I_l^s} y_i + C\frac{1}{|A/A_l^s|}\sum_{j \in I/I_l^s} z_j, \quad (38)$$

$$w'\left(a^i - a_s\right) + \xi\left\|a^i - a_s\right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l^s, \quad (39)$$

$$-w'\left(b_l^j - a_s\right) - \xi\left\|b_l^j - a_s\right\|_1 + \gamma + 1 \le z_j, \quad \forall j \in \frac{I}{I_l}, \quad (40)$$

$$y = \left(y_1, \ldots, y_{I_l^s}\right) \in R_+^{I_l^s}, \; w \in R^n, \; \xi \in R, \; \gamma \ge 1. \quad (41)$$

Table 1: The brief description of The Moods of Bloggers dataset.

Number of classes (moods)	4
Number of instances (blog posts)	157
Number of attributes (word stems)	23018
Mean number of word stems in blog posts	247

Let $w_s, \xi_s, \gamma_s, y_s$ be the solution of (P_l^s),

$$g_l^s(x) = g_{(w_s,\xi_s,\gamma_s,a_s)}(x). \tag{42}$$

Step 4. If $s < k$, let $s=s+1$, $A_l^s = \{a^i \in A_l^{s-1} : g_l^s(a^i) > 0\}$, $I_l^s = \{i \in I_l^s : a^i \in A_l^s\}$ and go to Step 3.

Step 5. If $l<c$, let $l=l+1$ and go to Step 1.

Step 6. Determine the function $g(x)$ parting A_l, $l=1,\ldots,c$, *as follows:*

$$g(x) = \min_l g_l^{1,\ldots,k}(x), \tag{43}$$

and stop.

As is seen, in the whole given algorithms, the linear programming subproblem includes inequality constraints (see (19), (20), (25), (26), (33), (34), (39), and (40)). These inequality constraints ensure classifying the text into the right category (class) by allowing misclassifications (y_i,z_j) as in (19), (25), (26), (33), (39), and (40). In inequalities of (20) and (34) constraints, no misclassifications are allowed by determining the $y_i=z_j=0$. While inequality constraints with ">0" ensure the data to be located outside of the obtained polyhedral conic function, inequality constraints with "< 0" ensure the data to fall into the obtained polyhedral conic function.

In the following section, given algorithms will be implemented on real-world text datasets for comparison with state-of-the-art methods and to verify the efficiency of PCF algorithms on large datasets.

6. Experiments

Primarily, to verify the efficiency of the PCF algorithms in text categorization, we benefit from a real-world dataset, "The Moods of Bloggers", that includes 157 blog posts written in four different moods, "cheerful, nervous, sad, and complicated" [48]. The attributes of the instances (feature vectors) are defined by the number of every word stem (w_i) existing in the document. That is to say, we study with a numerical dataset. The brief description of the dataset is given in Table 1. A desktop computer with Intel(R) Core(TM) i5-4460 CPU @ 3.20 GHz, 8 GB RAM, and 64-bit operating system is used in the experiments.

Algorithms 3 and 4 given in Section 4 were designed for binary classification so just to see how these algorithms work; we modified The Moods of Bloggers dataset as a binary dataset that includes two classes, "cheerful and others". As is seen, a single change is made in the number of classes. The implementations are made on MATLAB (multiparadigm numerical computing environment). The obtained results in terms of running times, accuracy, and F-measure are given in Table 2. Time shows the running time of the algorithm in seconds and accuracy value is determined as the ratio between the number of correct labeled points of the dataset and the number of the points in the whole dataset as follows [43]:

Table 2: Results for binary text classification on "The Moods of Bloggers".

	Algorithm 3	Algorithm 4
F-measure	1	1
Accuracy %	100	100
Time Sec.	200.91	37.86

cc: number of correct classified points of the dataset

te: number of instances of the dataset

$$\text{Accuracy} = \frac{100 * cc}{te}. \tag{44}$$

F-measure is the harmonic mean of precision and recall. Precision represents the proportion of predictive positive cases that are real positives and recall is the proportion of actual positive cases that were correctly predicted. These measures are presented as follows [49]:

$$\begin{aligned} \text{Precision} &= \frac{True\ positives}{True\ positives + False\ positives}, \\ \text{Recall} &= \frac{True\ positives}{True\ positives + False\ negatives}, \\ F-measure &= 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{45}$$

As is seen in Table 2, Algorithm 4 is more efficient than Algorithm 3 with regard to the running time. Clustering algorithm that is added to the initialization step decreases running time by reducing the step size that is required for finding the center points of polyhedral conic functions and correlatively number of solved linear programming subproblems. Accuracy value, %100, is obtained on both of the algorithms since PCF algorithm (Algorithm 3) ends after a finite number of iterations and the function $g : R^n \to R$ defined in the linear programming subproblem strictly separates the sets A and B. This theorem is proved in [8]. But it is clear that, according to the used dataset, obtained accuracy value in Algorithm 4 can be lower than Algorithm 3 because of using misclassifications for both of the classes.

Most of text categorization problems are multiclass classification problems; in other words, they are formed with more than two categories, so we utilize Algorithms 5 and 6 which are expressed in Section 5. As given in Table 1, The Moods of Bloggers dataset is suitable for these multiclass classification algorithms. Results obtained are given in Table 3.

Table 3: Results for multiclass text classification on "The Moods of Bloggers".

	Algorithm 5	Algorithm 6
Accuracy %	100	100
Time Sec.	155.30	145.119

As is seen in Table 3, Algorithms 5 and 6 are not so different from each other regarding accuracy and running time. Running times are close values since we use clustering algorithm in both of the methods.

We use training and testing terms in Tables 4 and 5 as performance metrics. Here, training term is the same as accuracy since we make training and testing on the same dataset. But testing term is a more reliable performance metric that we obtain by implementing cross-validation. We utilize tenfold cross-validation for a better comparison between PCFs and state-of-the-art methods. In tenfold cross-validation, 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 $t = \{1, 2, \ldots, 10\}$, it is trained on $D\backslash D_t$ and tested on D_t [50]. The presented testing value in Tables 4 and 5 is the mean value of 10 different accuracy values that is obtained by cross-validation. That is why the test results are not so high as in training results.

In Tables 4 and 5, respectively, for binary and multiclass classification, expressed algorithms are compared with the other state-of-the-art classification algorithms (Naive Bayes, classification via regression, J48 (decision tree)) by using WEKA (Waikato Environment for Knowledge Analysis), in terms of 10-fold cross-validation. In PCF algorithms, the best test values are obtained in Algorithms 4 and 6 since misclassifications for both classes are used in these algorithms. This constraint does not allow *overfitting* the problem. When we compare PCF algorithms with the others regarding test values, Algorithms 4 and 6 are more efficient than the other state-of-the-art methods except classification via regression.

Besides a detailed experiment on Moods of Bloggers dataset, we make implementations on real-world text datasets available in UCI (Machine Learning Repository). The datasets are represented by vectorial using and the attribute types are real or integer. Each attribute corresponds to a precise word or stem in the entire dataset vocabulary. TF_IDF formula is used as term weighting. These processes are expressed in detail in Section 2. The other details of used datasets are given in Table 6 and they are explained as follows.

Burst Header Packet (BHP). Burst Header Packet flooding attack on Optical Burst Switching (OBS) Network Data Set includes 1075 instances with 22 attributes. The last attribute stands for the classes as NB-No Block, Block, No Block, and NB-Wait [51].

CNAE-9. CNAE-9 dataset contains 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of 9 categories. This dataset is highly sparse (99.22% of the matrix is filled with zeros) [52].

Turkish Text Categorization (TTC). Turkish text categorization dataset is a collection of Turkish news and articles including categorized 3,600 documents from 6 well-known portals in Turkey [53].

DBWorld E-Mails. DBWorld e-mails dataset contains 64 e-mails which are manually collected from DBWorld mailing list. They are classified as "announces of conferences" and "everything else". Each attribute corresponds to a precise word or stem in the entire dataset vocabulary [54].

Obtained accuracy and time results are presented in Table 7. "-" is used for out of memory message in MATLAB. When we comment on the results we can say that Algorithms 5 and 6 are not so effective in terms of running times but it should not be forgotten that they are implied on MATLAB (a software environment) not in WEKA (a machine learning software). When we compare the accuracy results, we can say that Algorithm 5 is better than the others on composing good separator functions between classes.

7. Conclusion

In this paper, supervised classification via polyhedral conic functions is used to solve text classification problems. Binary and multiclass classification algorithms via PCFs are proposed and numerical experiments are done by implementing both of the proposed algorithms on a real-world dataset, called "The Moods of Bloggers". For performance metric, accuracy, running time, and tenfold cross-validation results are used. The obtained consequences are shown in tables. Besides, to augment the experiments and comparison with state-of-the-art methods, same work is done on four real-world text datasets available in UCI (Machine Learning Repository). If we comment on the results, we can say that classification algorithms via polyhedral conic functions are usable for text classification as well as other state-of-the-art algorithms. For future studies, these algorithms can be experienced by different structured text datasets on more effective software programs.

Data Availability

The real-world datasets supporting the conclusions of this article are available in the UCI repository [http://archive.ics.uci.edu/ml/index.php]. "The Moods of Bloggers" dataset supporting the conclusions of this article is available in Kemik Natural Language Processing Group Datasets [http://www.kemik.yildiz.edu.tr/?id=28].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors participated in every phase of research conducted for this paper. All authors read and approved the final manuscript.

Table 4: Results for 10-fold cross-validation of binary text classification on "The Moods of Bloggers".

	Algorithm 3	Algorithm 4	Naive Bayes	Classification via regression	J48
Training %	100	100	98	90	74
Testing %	65.83	75.45	71	78	74

Table 5: Results for 10-fold cross-validation of multiclass text classification on "The Moods of Bloggers".

	Algorithm 5	Algorithm 6	Naive Bayes	Classification via regression	J48
Training %	74.52	73.52	92	96	42
Testing %	45.12	49.56	43	50	42

Table 6: Details of real-world datasets.

	Burst Header Packet (BHP)	CNAE-9	Turkish text categorization (TTC)	DBWorld e-mails
Number of instances	1075	1080	3600	64
Number of attributes	22	857	3208	230
Number of classes	4	9	6	2

Table 7: Results for multiclass text classification on real-world datasets.

		Algorithm 5	Algorithm 6	Naive Bayes	Classification via regression	J48
BHP	Accuracy (%)	72.09	56.74	72.09	99.06	100
	Time (sec.)	29.08	76.50	0.02	0.01	0.04
CNAE-9	Accuracy (%)	100	99.69	96.13	70.02	94.41
	Time (sec.)	173.24	428.73	0.1	7.12	0.95
TTC	Accuracy (%)	-	-	87.75	87.75	93
	Time (sec.)	-	-	169.09	192.49	31.18
DBWorld e-mails	Accuracy (%)	100	98.43	98.43	87.5	92.18
	Time (sec.)	0.72	0.9	0.01	0.04	0.02

Acknowledgments

Dr. Burak Ordin acknowledges TUBITAK for its support (Project no. 113E763).

References

[1] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015.

[2] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," *Production and Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[3] D. L. Olson and D. D. Wu, "Data Mining Models and Enterprise Risk Management," in *Enterprise Risk Management Models*, Springer Texts in Business and Economics, pp. 119–132, Springer, Berlin, Germany, 2017.

[4] C. Romero and S. Ventura, "Data mining in education," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 12–27, 2013.

[5] P. Flach, *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*, Cambridge University Press, New York, NY, USA, 2012.

[6] T. Joachims, *Text categorization with support vector machines: learning with many relevant features*, Universit£t Dortmund lnformatik LS8, Baroper Str. 301, Germany, 1999.

[7] W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 879–886, 2008.

[8] R. N. Gasimov and G. Öztürk, "Separation via polyhedral conic functions," *Optimization Methods & Software*, vol. 21, no. 4, pp. 527–540, 2006.

[9] S. H. Eui-Hong, K. George, and K. Vipin, *Text Categorization Using Weighted Adjusted k-Nearest Neighbor Classification*, Department of Computer Science and Engineering. Army HPC Research Centre, University of Minnesota, Minneapolis, USA, 1999.

[10] I. Hmeidi, B. Hawashin, and E. El-Qawasmeh, "Performance of KNN and SVM classifiers on full word Arabic articles," *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 106–111, 2008.

[11] V. Tam, A. Santoso, and R. Setiono, "A comparative study of centroid-based, neighborhood-based and statistical approaches

for effective document categorization," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 235–238, 2002.

[12] S. L. Bang, J. D. Yang, and H. J. Yang, "Hierarchical document categorization with k-NN and concept-based thesauri," *Information Processing & Management*, vol. 42, no. 2, pp. 387–406, 2006.

[13] R. Alhutaish and N. Omar, "Arabic text classification using K-nearest neighbour algorithm," *International Arab Journal of Information Technolog*, vol. 12, no. 2, pp. 190–195, 2015.

[14] J. Rocchio, "Relevance Feedback in Information Retrieval," in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Salton, Ed., Chapter 4, pp. 313–323, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

[15] D. Ittner, D. Lewis, and D. Ahn, "Text Categorization of Low Quality Images," in *Symposium on Document Analysis and Information Retrieval*, pp. 301–315, Las Vegas, NV, USA, 1995.

[16] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[17] M. Pazzani and D. Billsus, "Learning and revising user profiles: the identification of interesting web sites," *Machine Learning*, vol. 27, no. 3, pp. 313–331, 1997.

[18] A. Zeng and Y. Huang, "A text classification algorithm based on rocchio and hierarchical clustering, advanced intelligent computing," in *7th international conference, ICIC 2011*, pp. 432–439, Zhengzhou, China, 2011.

[19] A. McCallum and K. Nigam, "A comparison of event models for naïve bayes text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.

[20] R. Irina, "An empirical study of the naïve bayes classifier," in *Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[21] R. Irina, H. Joseph, and T. Jayram, *An Analysia of Data Characteristics that affect Naïve Bayes Performance*, IBM T. J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532, USA, 2001.

[22] W. Miah, J. Yearwood, and S. Kulkarni, "Detection of child exploiting chats from a mixed chat dataset as a text classification task, Conference: Australasian Language Technology Association Workshop," December 2011.

[23] J. W. Kim, B. H. Lee, M. J. Shaw, H.-L. Chang, and M. Nelson, "Application of decision-tree induction techniques to personalized advertisements on internet storefronts," *International Journal of Electronic Commerce*, vol. 5, no. 3, pp. 45–62, 2001.

[24] R. Greiner and J. Schaffer, *AIxploratorium – Decision Trees*, Department of Computing Science, University of Alberta, Edmonton, ABT6G2H1, Canada, 2001.

[25] A. Mammone, M. Turchi, and N. Cristianini, "Support Vector Machines," *Wires: Wiley's Interdisciplinary Reviews in Computational Statistics*, vol. 1, no. 3, pp. 283–289, 2009.

[26] R. Luss and A. D'Aspremont, "Predicting abnormal returns from news using text classification," *Quantitative Finance*, vol. 15, no. 6, pp. 999–1012, 2015.

[27] K. Fragos, P. Belsis, and C. Skourlas, "Combining Probabilistic Classifiers for Text Classification," *rocedia - Social and Behavioral Sciences, Volume 147 Pages 307–312, 3rd International Conference on Integrated Information (IC-ININFO)*, vol. 147, pp. 307–312, 2014.

[28] S. Keretna, C. P. Lim, D. Creighton, and K. B. Shaban, "Classification ensemble to improve medical named entity recognition," in *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2014*, San Diego, CA, USA, 2014.

[29] A. Jain and J. Mandowara, "Text classification by combining text classifiers to improve the efficiency of classification," *International Journal of Computer Application*, vol. 6, no. 2, 2016.

[30] A. H. Aliwy and Ameer. E. H. A., "Comparative study of five text classification algorithms with their improvements," *International Journal of Applied Engineering Research ISSN 0973-4562*, vol. 12, no. 14, pp. 4309–4319, 2017.

[31] M. M. Mironczuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36–54, 2018.

[32] C. C. Aggarwal, "Mining Text Data," in *Data Mining*, pp. 429–455, Springer, Boston, MA, USA, 2015.

[33] C. C. Aggarwal and C. A. Zhai, "Survey of Text Classification Algorithms," in *Mining Text Data*, Springer, Boston, MA, USA, 2012.

[34] G. M. D. Nunzio, "A new decision to take for cost-sensitive Naïve Bayes classifiers," *Information Processing & Management*, vol. 50, no. 5, pp. 653–674, 2014.

[35] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, Part B, pp. 806–814, 2016.

[36] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to Spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206–10222, 2009.

[37] Bhumika, Sehra. S, and A. Nayyar, "A review paper on algorithms used for text classification," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 3, 2013.

[38] A. M. Bagirov, "Max-min separability," *Optimization Methods & Software*, vol. 20, no. 2-3, pp. 277–296, 2005.

[39] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, no. 1, pp. 23–34, 1992.

[40] A. Astorino and M. Gaudioso, "Polyhedral Separability Through Successive LP," *Journal of Optimization Theory and Applications*, vol. 112, no. 2, pp. 265–293, 2002.

[41] G. Öztürk, A. M. Bagirov, and R. Kasimbeyli, "An incremental piecewise linear classifier based on polyhedral conic separation," *Machine Learning*, vol. 101, no. 1-3, pp. 397–413, 2014.

[42] N. Uylas, *Methods based on mathematical optimization for data classification*, Ege University, 2013.

[43] N. U. Sati, "A binary classification algorithm based on polyhedral conic functions," *Düzce University Journal of Science and Technology*, vol. 3, pp. 152–161, 2015.

[44] A. Kusiak, "Data analysis: models and alogrithms," in *Proceedings of the SPIE 4191, Sensors and Controls for Intelligent Manufacturing*, 2001.

[45] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, NY, USA, 1973.

[46] L. Rokach and O. Maimon, *Clustering Methods, Data Mining and Knowledge Discovery Handbook*, Chapter 15, 2005.

[47] G. Öztürk and M. T. Çitfçi, "Clustering based polyhedral conic functions algorithm in classification," *Journal of Industrial and Management Optimization*, vol. 11, no. 3, pp. 921–932, 2015.

[48] Kemik Doğal Dil İşleme Grubu, http://www.kemik.yildiz.edu.tr/ (2009), Date accessed: March, 2017.

[49] M. D. P. Salas-Zárate, R. Valencia-García, A. Ruiz-Martínez, and R. Colomo-Palacios, "Feature-based opinion mining in financial news: an ontology-driven approach," *Journal of Information Science*, 2016.

[50] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the International Joint Conference on Artificial Intelligence*, San Francisco, 1137, Cal, USA, 1995.

[51] A. Rajab, C. Huang, M. Al-Shargabi, and J. Cobb, "Countering burst header packet flooding attack in optical burst switching network," in *International Conference on Information Security Practice and Experience*, pp. 315–329, Springer International Publishing, 2016.

[52] P. M. Ciarelli and E. Oliveira, "Agglomeration and elimination of terms for dimensionality reduction," in *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications*, pp. 547–552, December 2009.

[53] D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, and E. Borandag, "TTC-3600: A new benchmark dataset for Turkish text categorization," *Journal of Information Science*, Published online before print December 29, pp. 174–185, 2015.

[54] M. Filannino, "DBWorld e-mail classification using a very small corpus," in *Project of Machine Learning course*, University of Manchester, 2011.