Turkish Journal of Electrical Engineering & Computer Sciences

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

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 80 – 92
© TÜBİTAK
doi:10.3906/elk-1905-45

A novel semisupervised classification method via membership and polyhedral conic functions

Nur UYLAŞ SATI*
Bodrum Maritime Vocational School, Muğla Sıtkı Koçman University, Bodrum, Muğla, Turkey

Received: 08.05.2019 • **Accepted/Published Online:** 18.10.2019 • **Final Version:** 27.01.2020

Abstract: In real-world problems, finding sufficient labeled data for defining classification rules is very difficult. This paper suggests a new semisupervised multiclass classification method. In the initialization, new membership functions are defined by utilizing the labeled data's medoids and means. Then the unlabeled points are labeled with the class of the highest membership value. In the supervised learning phase, separation via the polyhedral conic functions (PCFs) approach is improved by using defined membership values in the linear programming problem. The suggested algorithm is tested on real-world datasets and compared with the state-of-the-art semisupervised methods. The results obtained indicate that the suggested algorithm is effective in classification and is worth studying.

Key words: Semisupervised classification, multiclass classification, membership functions, polyhedral conic functions

1. Introduction

Data mining is one of the fastest growing research areas thanks to the increase in registrable and accessible data by the recent developments in technology. Data classification is a design of a definite recognition system that uses a training set selected from a determined dataset. It has unsupervised, semisupervised, and supervised learning types according to the determined data's labels. If all data are unlabeled, it is called unsupervised learning. Supervised learning, on the other hand, uses only labeled data. In semisupervised learning, there are both labeled and unlabeled data and it can be positioned halfway between unsupervised and supervised learning models. Since it is a very expensive and time-consuming process to label data, most of the real-world datasets consist of both labeled and unlabeled data. Therefore, in the present paper, we study semisupervised learning. This learning type can be used in various applications like spam filtering, speech recognition, text classification, video regulation, image categorization, protein 3D structure prediction, web content classification, and students' performance prediction.

Semisupervised classification techniques can be thought of as a combination of unsupervised and supervised techniques and it is aimed to utilize both of them. Much used semisupervised learning methods have different types, called self-training, cotraining, transductive support vector machines, and graph-based methods.

Self-training is a droning method that uses more data in every phase. Firstly, it builds a classifier by using just labeled data; then iteratively unlabeled data with a labeling confidence exceeding a certain threshold after being classified by the defined classifier are added to the process [1].

Starting with a set of labeled data, cotraining algorithms attempt to increase the amount of annotated data using some (large) amounts of unlabeled data like self-training. However, cotraining uses several classifiers

*Correspondence: nuruylas@gmail.com

differently from self-training and these classifiers are used to label the unlabeled ones. The most confident ones are added to the labeled data and the process continues iteratively [2].

Transductive support vector machines (SVMs) perform initial learning by using inductive learning on all labeled examples and generate an initial SVM classifier. By using the obtained classifier, a specified number of N unlabeled examples are labeled. A temporary effect factor is set. The SVM is retrained over all the examples. The objective function of the SVM is tried to be decreased by label switching. The value of the temporary effect factor is increased slightly and the same processes after retraining the SVM over all the examples are executed. The algorithm goes on until the temporary factor is equal or bigger than the defined effect factor in the initialization [3].

In the literature, the most recent papers deal with semisupervised learning in different areas such as text sentiment classification, grading multiple peer-reviewed open-ended works, and article selection for medical systematic reviews [4–8].

Based on the idea of effectively utilizing unlabeled samples, a synthetic framework that covers the whole process of semisupervised learning from seed selection and iterative modification of the training text set to the cotraining strategy of the classifier is proposed for text sentiment classification [4].

A novel methodology for grading multiple peer-reviewed open-ended works is proposed [5]. In the analysis, statistical semisupervised algorithms are used to detect possible biased scorings. This methodology has been tested on two different assignments with two heterogeneous groups of people to assess the robustness and reliability of the proposal.

Semisupervised approaches are used for systematic reviews [6]. Unlabeled documents are utilized by propagating labels in a close neighborhood and the space of spectral embeddings is used for better distance representation.

A solution for selecting the most influential label based on using the relations among the labels and features to a semisupervised multilabel classification algorithm on texts is proposed [7]. Experiments are done on two datasets of Vietnamese reviews and English emails of Enron and the results show the positive effects of the proposal.

A cluster-then-label method is proposed to identify high-density regions in the data space, which are then used to help a supervised SVM in finding the decision boundary [8]. The suggested method is compared with other supervised and semisupervised state-of-the-art techniques using two different classification tasks applied to breast pathology datasets.

Data classification also has types according to the number of classes. If the dataset consists of two different classes (labels), performing workpiece is called binary classification. If it has more than two different classes (labels) it is called multiclass or multilabel classification. In the literature, many studies can be found on multiclass classification currently [9–14]. Previous studies [12–14] focus on both semisupervised and multiclass classification.

In this paper, differently from the proposed approaches on semisupervised learning in the literature, membership functions and polyhedral conic functions are utilized together to improve the classification performance. A novel multiclass semisupervised classification method is suggested. In the initialization, differently from inductive learning and classifier generation, we define two membership functions by using both the medoids and k-means of labeled points. Every unlabeled point is computed on one of the defined functions and membership values are obtained for each class. These membership values are used firstly in labeling the unlabeled points and later on in the linear programming problem of supervised learning via polyhedral conic functions.

The rest of the paper is organized as follows. In the methodology's first subsection, the membership approach used is discussed and the membership functions are presented by defining equations. Polyhedral conic functions, used in the supervised learning phase, are explained in the second subsection. In the third subsection of the methodology, the suggested semisupervised multiclass classification algorithm is given. In the third section, numerical experiments are performed by implementing the suggested algorithm on real-world datasets. The suggested algorithm is compared with the state-of-the-art algorithms in terms of accuracy results. In the fourth section, a discussion chapter is presented for explaining and evaluating the findings and making an argument in support of the overall conclusion. Finally, in the last section, the paper is concluded and the future studies are explained.

2. Methodology

2.1. Membership functions

In this paper, a novel semisupervised multiclass classification method is defined for problems with very few labeled and many unlabeled data. Separation via the polyhedral conic functions approach is utilized in the supervised learning phase. Before the supervised learning phase, labeling the unlabeled points by using membership values is aimed. For defining the membership values of the points, two new membership functions are defined by utilizing the medoids of the classes and *k*-means.

Membership functions were first introduced in 1965 by Lofti A. Zadeh for fuzzy sets. A fuzzy set is defined as a class of objects with a given grade of membership. Such a set is characterized by a membership (characteristic) function that assigns to each object a grade of membership ranging between zero and one [15]. In semisupervised problems, the unlabeled points can form a fuzzy set. The fuzziness of a fuzzy set is defined by a membership function. The shapes of the membership functions are important for particular problems because they can affect the fuzzy inference systems. In the literature, many membership functions are defined for particular problems [16].

In the present paper, we are not dealing with particular problems and we have labeled points that we can utilize and so firstly a general membership function is defined by utilizing the medoids of these labeled points. This membership function is more effective when each of the classes is structured in one piece.

Medoids is the term used for the points whose sum of the distance to all the objects in the same cluster is minimal. Medoids are similar in concept to means or centroids, but medoids are always restricted to being members of the dataset. The medoids are obtained from the labeled points as follows:

$$x_{l-_{medoid}} = argmin_{y\in\left\{x_1,x_2,\ldots,x_{|I_l|}\right\}} \sum_{i=1}^{|I_l|} d(y,x_i). \tag{1}$$

Here l represents the class and I_l is the number of the labeled points in l class. *d(y,x_i)* is the euclidian distance between y and x_i points. After obtaining the medoids of each class the membership function is defined as follows:

$$\delta_{ik} = \begin{cases} \frac{\sum_{l=1}^{s} d(x_{l-medoid}-x_i)-d(x_{k-medoid}-x_i)}{(s-1)*\sum_{l=1}^{s} d(x_{l-medoid}-x_i)}, & \text{for } x_i \in C, \\ 1, & \text{for } x_i \in A_k, \\ 0, & \text{for } x_i \notin A_k \vee x_i \notin C. \end{cases} \tag{2}$$

Here "s" is defined for the number of the classes, "δ_{ik}" is the membership value of $x_i$for the kth class, $A_l$represents the labeled classes for l=1,2,...,s, and C is the set of unlabeled points.

The membership function values of the original labeled points are determined as “1” for the classes they are labeled and “0” for the other classes. The unlabeled ones’ membership values are determined by using the equation for $x_i \in\ C\ in$ (2).

If each labeled class has a piecewise structure, the k-means method can be applied to determine the real structure of the dataset. In this case, after finding the j numbers of center points of each class by k-means algorithm on the labeled dataset, the membership function is determined as follows:

$$\delta_{ik} = \begin{cases} \frac{\sum_{l=1}^{s}(\min\limits_{j} d(x_i,l_j)) - \min\limits_{j} d(x_i,k_j)}{\sum_{l=1}^{s}(\sum_{l=1}^{s}(\min\limits_{j} d(x_i,l_j)) - \min\limits_{j} d(x_i,l_j))}, & \text{for } x_i \in C \text{ and } l_j = \text{the } jth \text{ center of ''}l\text{'' class}, \\ \\ 1, & \text{for } x_i \in A_k, \\ 0, & \text{for } x_i \notin A_k \vee x_i \notin C. \end{cases} \tag{3}$$

Due to the membership functions’ features, the total membership value of a datum point for all clusters must be unity:

$$for\ \forall i,\ i = 1, ..., m, \sum_{k=1}^{s} \delta_{ik} = 1 \tag{4}$$

and defined membership functions (2) and (3) both obey this rule.

After obtaining the membership values of unlabeled points, we label them with the class number of the biggest membership value. Supervised learning is processed after these initialization studies. The defined membership function is also used in the objective function of the linear programming problem in the supervised learning phase. The aim is to regulate the effect of the data in finding the separation rule. The nonoriginal labeled points used in the training phase should have a smaller effect on the learning process.

In the present paper, the membership function defined in (2) is used in the suggested algorithm. However, in the numerical experiments for comparative analysis, the membership function defined in (3) is also tested.

2.2. Polyhedral conic functions (PCFs)

PCFs were firstly defined by Gasimov and Öztürk in 2006 for separating two different datasets in R^n. The definition of PCFs is given as follows [17]:

Definition 1 *A function g: $R^n \ \rightarrow\ R$ is called polyhedral conic if its graph is a cone and all its sublevel sets $S\ \ \alpha\ = x \in\ R^n\ :\ g(x)\ \leq\ \ \alpha \}$, for $\alpha\ \in\ R$, are polyhedrons.*

Here the PCF $g_{(w,\xi,\gamma,a)}$:$R^n \ \rightarrow\ R$ is defined as

$$g_{(w,\xi,\gamma,a)} : R^n \rightarrow R = w'(x-a) + \xi \left\| x-a \right\|_1 - \gamma \tag{5}$$

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

Gasimov and Öztürk used PCFs in a linear programming problem to separate two disjoint sets and define the first PCF algorithm as follows [17]:

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

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

Algorithm 1: 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$_l$).

$$(P_l) \quad \min(\frac{y^{'} e_m}{m}), \tag{6}$$

$$w^{'}(a^i - a^l) + \xi \left\|a^i - a^l\right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l, \tag{7}$$

$$-w^{'}(b^j - a^l) - \xi \left\|b^j - a^l\right\|_1 + \gamma + 1 \le 0, \quad \forall j \in J, \tag{8}$$

$$y = (y_1, ..., y_m) \in R^m_+, w \in R^n, \xi \in R, \gamma \ge 1 \quad .$$

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

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

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

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

$$g(x) = \min_l g_l(x), \tag{9}$$

and stop.

This algorithm separates the sets strictly after sufficient iterations. However, this algorithm has a high time complexity because of using arbitrary points in step 1. To analyze this problem the algorithm is modified in the literature [18–20]. A clustering algorithm is suggested to find the vertex points of PCFs instead of testing arbitrary points and effective results are obtained in terms of computational time.

In defined Algorithm 1, another problem is overfitting because of a strict separation in datasets; this problem is solved in the literature [18–20] by adding misclassifications to the constraint given in (8) and adding these misclassifications to the objective function (2).

Algorithm 1 is formed for binary classification problems. Since most of the real-world problems have more than two classes, it is modified [20] for solving multiclass supervised data classification problems. For multiclass classification, Algorithm 1 is applied between each class and the rest. Designed multiclass classification algorithms are tested for text classification and good comparative results are obtained against the state-of-the-art text classification methods [21]. The mentioned PCF multiclass classification algorithm using both clustering and misclassifications is given as follows [20]:

Algorithm 2: Multiclass classification algorithm via clustering and PCFs

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

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

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

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

$$(P_l^s) \qquad \min \frac{1}{|A_l^s|} \sum_{i \in I_l^s} y_i + C \frac{1}{|A/A_l^s|} \sum_{j \in I/I_l^s} z_j, \tag{10}$$

$$w^{'}(a^i - a_s) + \xi \left\| a^i - a_s \right\|_1 - \gamma + 1 \leq y_i, \quad \forall i \in I_l^s, \tag{11}$$

$$-w^{'}(b_l^j - a_s) - \xi \left\| b_l^j - a_s \right\|_1 + \gamma + 1 \leq z_j, \quad \forall j \in I/I_l, \tag{12}$$

$$y = (y_1, ..., y_{I_l^s}) \in R_+^{I_l^s}, w \in R^n, \xi \in R, \gamma \geq 1 \ .$$

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

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

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,...,c as follows:*

$$g(x) = \min_l g_l^{1,...,k}(x),$$

and Stop.

We modify this multiclass supervised data classification algorithm for use in the supervised learning phase of the suggested semisupervised multiclass classification algorithm. The modification is discussed in the next section.

2.3. Semisupervised multiclass classification algorithm

The suggested semisupervised multiclass classification algorithm is given as follows:

Algorithm 3: Semisupervised Multiclass Classification Algorithm via membership and polyhedral conic functions

Let $A = A_1 \cup A_2 \cup ... \cup A_s, A = \{a_l^i \in IR^n : i \in I_l, l = 1, 2, ..., s\}$. Let C be a given unlabeled dataset containing $t \in Z^+$ *n*-dimensional vector:

$C = \{c^t \in R^n, t \in T\}$, where $T = \{1, ..., f\}$.

Initialization: Find the medoid points of each A_l as follows:

$$x_{l-_{medoid}} = argmin_{y \in \left\{x_1, x_2, ..., x_{|I_l|}\right\}} \sum_{i=1}^{|I_l|} d(y, x_i) \tag{13}$$

$x_{l-medoid} = a_l$, *l*=1,2,..,s. Let *l*=1.

Step 1: Define δ_{il} membership function for each $x_i \in A \bigcup C$ point for the class of A_l as follows:

$$\delta_{il} = \begin{cases} \frac{\sum_{k=1}^{s} d(x_{k-medoid} - x_i) - d(x_{l-medoid} - x_i)}{(s-1) * \sum_{k=1}^{s} d(x_{k-medoid} - x_i)}, & \text{for } x_i \in C, \\ \\ 1, & \text{for } x_i \in A_l, \\ 0, & \text{for } x_i \notin A_l \vee x_i \notin C. \end{cases}$$

Step 2: For all unlabeled data $c_i \in C, i = 1, ..., f$, let $c_i \in A_l$, *where* $l = \arg\max \delta_{il}$, $l = 1, ..., s.$.

Step 3: Update the dataset with the new labeled data. Let *l*=1.

Step 4: $B = A/A_l, B = \{b_l^j \in R^n : j \in I/I_l\}$.

Step 5: Apply clustering algorithm in A_l. Let *cs* be the number of clusters and *ks*=1, $I_l^1 = I_l$, and $A_l^1 = A_l$.

Step 6: Let $a_{ks} \in A_l^{ks}$ be the *ks*th center of A_l. Solve P_l^{ks} subproblem.

$$(P_l^{ks}) \quad \min(\frac{1}{|A_l^{ks}|} \sum_{i=1}^{|I_l^{ks}|} \delta_{il} y_i), \tag{14}$$

$$w^{'}(a^i - a_{ks}) + \xi \left\|a^i - a_{ks}\right\|_1 - \gamma + 1 \leq y_i, \quad \forall i \in I_l^{ks}, \tag{15}$$

$$-w^{'}(b_l^j - a_{ks}) - \xi \left\|b_l^j - a_{ks}\right\|_1 + \gamma + 1 \leq 0, \quad \forall j \in I/I_l, \tag{16}$$

$$y = (y_1, ..., y_{|I_l^{ks}|}) \in R_+^{I_l^{ks}}, w \in R^n, \xi \in R, \gamma \geq 1 \quad .$$

Let $w_l, \xi_l, \gamma_l, y_l$ be the solution of (P_l),

$$g_l^{ks}(x) = g_{(w_l^{ks}, \xi_l^{ks}, \gamma_l^{ks}, a_l^{ks})}(x).$$

Step 7: If $ks < cs$, let *ks*=*ks*+1, $A_l^{ks} = \{a^i \in A_l^{ks-1} : g_l^{ks}(a^i) > 0\}, I_l^{ks} = \{i \in I_l^{ks} : a^i \in A_l^{ks}\}$and go to Step 6.

Step 8: If *l*< *s,* Let *l*=*l*+1 and go to Step 9.

Step 9: Determine the function *g(x)* parting A_l, *l=1,...,s* as follows:

$$g(x) = \min_l g_l^{1,...,cs}(x),$$

and Stop.

In the suggested semisupervised multiclass classification algorithm's initialization phase, the medoids of the classes are found by using the labeled points and applying equation (13). In steps 1 and 2 after calculating the membership function given in (2) for each datum, the unlabeled data are assigned to the class (l) of the biggest membership value (δ_{il}). In step 3, the dataset is updated and in step 4 the supervised learning phase starts.

In the supervised learning phase, Algorithm 2 given in subsection 2.2 is utilized. Differently from Algorithm 2, membership values are used in the objective function of the linear programming subproblem (14). In the linear programming subproblem, the aim is to define a separative function with minimum error as shown in (14). Some of the points in the class to be separated are original labeled and some are nonoriginal since they are labeled with the biggest membership valued class. To obtain an effective separative function, the error values of each point are multiplied by their membership values in the objective function (14). Since the original labeled points' membership values are determined as "1" and the others are determined by given equation (2), the effect of nonoriginal labeled points on the separative function is decreased. The second constraint (16) in the linear programming problem is modified by not allowing misclassifications and defined as the constraint (8) in Algorithm 1. This is because we already prevent overfitting by using nonoriginal labeled data in training.

The initialization and step 1 of the above given Algorithm 3 can be changed by using membership function (3) according to the structure of the labeled dataset as mentioned in section 2.1.

3. Numerical experiments

In numerical experiments, to understand the effect of the membership values used in the training phase of Algorithm 3, we also apply Algorithm 3 without membership values in the objective function (14). This algorithm is called "Algorithm3WMV" (without membership value). For performance measurement, testing accuracy values, which are the ratio between the number of the correctly labeled data and the number of the data in the whole dataset, are used as follows:

cc: number of correctly classified points of the dataset

te: number of points in the dataset

$$accuracy = \frac{100 * cc}{te}$$

In the experiments, six real-world datasets called Iris, Cleveland, Wine, Vehicle, Yeast, and Haberman are used since they are multiclassed and labeled datasets. The datasets are obtained from the UCI machine learning repository [22]. These datasets are frequently used in supervised classification research as they are labeled datasets. In the present paper, since we are dealing with semisupervised classification, we use 30% of the dataset as labeled and 70% of the dataset as unlabeled. In the testing phase, the whole (100%) labeled dataset is used. The properties of the datasets are given in Table 1. A desktop computer with Intel Core i5-4460 CPU @ 3.20 GHz, 8 GB RAM, and 64-bit operating system is used in the experiments.

Table 1. Brief description of the datasets.

Name of the dataset	No. of attributes	No. of classes	No. of instances
Iris	4	3	150
Cleveland	13	4	297
Wine	13	3	178
Vehicle	18	4	946
Yeast	8	10	1484
Haberman	3	2	306

For presenting comparative results with the state-of-the-art semisupervised methods, along with suggested Algorithm 3 and Algorithm3WMV, by using the WEKA Tool, Yatsi, LLGC, Weight, and CollectiveWrapper and by using the KEEL Tool, Self training (LMT), Self training (C45), Self training (NN), CoBag (C45), Cotrain (C45), Cotrain (SMO), Demogratic-co, Tri-train (C45), Tri-train (SMO), and SETRED algorithms are applied to the same training and testing datasets. Brief descriptions of these methods are given as follows:

Yatsi: "Yet Another Two Stage Idea" uses the given classifier to train on the training set and label the unlabeled data. In our experiments, J48 is chosen as a classifier. Predictions are then done via nearest-neighbor-search and a majority over the *k* nearest neighbors (actually their weights) of the instance that is to be predicted (in case of ties the first label is chosen) [23, 24].

LLGC: "Learning with local and global consistency" was presented in 2003 by Zhou et al. [25]. It is a collective classifier and it generates a smooth classifier function for labeled and unlabeled data. In our experiments euclidian distance is used as distance metric [23].

Weight: It is a collective classifier that uses one classifier for labeling the test data after training on the train set. The trained classifier determines the class labels for all the instances in the test dataset. This is again input for another classifier. In the initializing step, all instances from the test set have a weight of 0. In each following step, they get a weight of current_step / number_of_steps. This implies that all provided classifiers need to be able to handle weighted instances [26].

CollectiveWrapper: It represents a wrapper around any normal WEKA classifier. In our experiments, Naive Bayes is chosen as a classifier [26].

Self training (LMT): It combines a self-training scheme with a logistic model tree (LMT) algorithm. A LMT is a decision tree that has linear regression models at its leaves to provide a piecewise linear regression model [27].

Self training (C45): It combines a self-training scheme with the C45 algorithm. The C45 algorithm is used as a decision tree classifier that can be employed to generate a decision, based on a certain sample of data [28].

Self training (NN): In this algorithm, the k-nearest neighbors (k-NN) classifier was chosen in making a decision during the self-training process. k-NN is a supervised learning algorithm where the result of a new instance query is classified based on a majority of k-nearest neighbor category [29].

CoBag (C45): Cobagging is an ensemble method that combines the predictions from many machine learning algorithms to make more reliable and accurate predictions than any individual model. CoBag (C45) is a standard cobagging method using the C45 method as base classifier during self-labeling as well as final classifier [30].

Cotrain (C45): Cotraining is a special case of the more general multiview learning. It combines a cotraining scheme with the C45 algorithm [31].

Cotrain (SMO): It combines a cotraining scheme with sequential minimal optimization (SMO). SMO is an algorithm for solving the quadratic programming (QP) problem that arises during the training of SVMs [2].

Demogratic-co: In demogratic colearning, multiple algorithms instead of multiple views enable learners to label data for each other. The technique leverages off the fact that different learning algorithms have different inductive biases and that better predictions can be made by the voted majority [32].

Tri-train (C45): It combines a tri-training scheme with the C45 algorithm. Tri-training is a style of cotraining that uses three classifiers to exploit unlabeled data [6]. These classifiers are then refined using unlabeled examples in the tri-training process [33].

Tri-Train (SMO): It combines a tri-training scheme with the SMO algorithm [33].

SETRED: It utilizes a specific data editing method to identify and remove the mislabeled examples from the self-labeled data. In detail, in each iteration of the self-training process, the local cut edge weight statistic is used to help estimate whether a newly labeled example is reliable or not, and only the reliable self-labeled examples are used to enlarge the labeled training set [34].

The state-of-the-art algorithms are implemented using WEKA (Waikato Environment Knowledge Analysis) and KEEL (Knowledge Extraction based on Evolutionary Learning) tools with ready to use codes. The defined and suggested Algorithm 3 codes are written and implemented on MATLAB. Algorithm 3 is applied twice both with two defined membership functions (2) and (3). The membership function (3) is applied twice both with using 2-means and 3-means algorithms. The results obtained are presented in Table 2 in terms of accuracy values. “-“ is used for out of memory messages in MATLAB.

Table 2. Accuracy values of implementations.

	Iris	Cleveland	Wine	Vehicle	Yeast	Haberman
Algorithm 3 with membership function (2)	92.66%	71.04%	79.21%	78.83%	-	76.75%
Algorithm 3 with membership function (3) and k=2	96.33%	64.30%	55.61%	78.70%	-	70.88%
Algorithm 3 with membership function (3) and k=3	92.66%	69.02%	79.77%	78.88%	-	76.74%
WMV	92%	69.02%	79.77%	78.73%	-	76.47%
Yatsi	96%	53.19%	53.19%	67.22%	54.76%	70.56%
LLGC	54%	37.03%	37.03%	24.15%	54.76%	71.02%
Weighting	96%	61.95%	61.95%	65.37%	52.16%	70.09%
Wrapper	96%	56.56%	56.56%	24.15%	55.14%	71.02%
SelfTraining LMT	89%	58.37%	94.38%	78.74%	58.43%	72.19%
Self Training NN	91.33%	51.59%	93.82%	66.30%	50%	66.30%
Self Training C45	92%	53.64%	84.15%	65.84%	52.09%	71.53%
CoBag C45	90.67%	55.99%	88.17%	65.61%	52.36%	72.82%
CoTrain C45	91.33%	54.67%	82.55%	66.56%	53.58%	72.88%
CoTrain SMO	94.67%	47.33%	96.63%	71.75%	52.70%	66.95%
Democ-Co	95.33%	57%	96.60%	56.52%	53.71%	74.47%
TriTrain C45	91.33%	53.62%	90.39%	67.02%	54.05%	70.89%
TriTrain SMO	94.67%	46.19%	95.52%	72.45%	52.44%	64.99%
SETRED	92.67%	55.66%	92.75%	65.83%	51.28%	67.61%

4. Discussion

When the implementations' results in Table 2 are examined, it is seen that the proposed Algorithm 3 is more effective than the state-of-the-art semisupervised methods in most cases.

In Cleveland and Haberman implementations, the best results are obtained by Algorithm 3 using membership function via medoids (2) since the defined classes in these datasets are formed in one piece as we have mentioned in the methodology section. In Iris and Vehicle implementations, using the k-means method in membership functions (3) is more effective since the defined classes in these datasets have a partial form. In the k-means method, the number of clusters (k) determined before the implementations is also very important such that in Iris and Vehicle implementations the best results are obtained when k is determined as 2 and 3, respectively.

In Wine implementations, Algorithm 3 is not effective when compared to KEEL implementations. In Yeast implementations, Algorithm 3 gets an “out of memory” message in MATLAB. In the implementations of Algorithm 3, better code writing and using a more powerful computer can increase the efficiency of the algorithm for large datasets since it can decrease the running times.

Algorithm 3 without membership values in the objective function (14), which is called ”Algorithm3WMV”, also gets good results in the experiments. The difference between Algorithm3WMV and Algorithm 3 accuracy results is not so high since the membership values of unlabeled data used in constraint (14) are so close to 1.

In general, the overall performance of Algorithm 3 is quite effective. The classification accuracy results obtained verify that our method is quite effective and indicate large potential for some real-life applications. Moreover, our approach provides a novel angle to study polyhedral conic functions in semisupervised learning and sheds some light on future research on approximation for membership values of unlabeled points. For future research, the defined Algorithm 3 can be modified by different membership functions or different weightings and it can be applied to various datasets using different software platforms and more powerful computers.

5. Conclusion

Semisupervised learning for multiclass classification is the subject of this paper. A novel algorithm is suggested using membership functions and polyhedral conic functions. Two membership functions are defined for both using in labeling unlabeled points and regulating the effects of unlabeled points in the training phase. One of them uses medoids (2) and one of them uses *k*-means (3) methods. Numerical experiments are done on the suggested algorithms using the MATLAB numerical analysis environment. Furthermore, the state-of-the-art semisupervised algorithms are applied for comparative analysis in terms of accuracy values. The suggested Algorithm 3 is effective in semisupervised multiclass classification since it gets higher accuracy values than others in most cases.

When a general conclusion is made in the light of these findings, it can be stated that approaches using PCFs in semisupervised learning can yield effective results. Also based on the suggested algorithm, different membership functions or different weightings can be used and applied to various datasets for future research.

References

[1] Mihalcea R. Co-training and self-training for word sense disambiguation. In: Proceedings of the 8th Conference on Computational Natural Language Learning; Boston, MA, USA; 2004. pp. 33-40.

[2] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory; Madison, WI, USA; 1998. pp. 92-100.

[3] Chen Y, Wang G, Dong S. Learning with progressive transductive support vector machine. Pattern Recognition Letters 2003; 24: 1845-1855.

[4] Li Y, Liv Y, Wang S, Liang J, Li J, Li X. Cooperative hybrid semi-supervised learning for text sentiment classification. Symmetry 2019; 11 (2): 133. doi: 10.3390/sym11020133

[5] Juan JRR, Javier Gallego A, Mas JJV, Zaragoza JC. Statistical semi-supervised system for grading multiple peer-reviewed open-ended works. Computers and Education 2018; 126: 264-282.

[6] Kontonatsios G, Brockmeier AJ, Przybya P, McNaught J, Mu T et al. A semi-supervised approach using label propagation to support citation screening. Journal of Biomedical Informatics 2017; 72: 67-76.

[7] Ha QT, Pham TN, Nguyen VQ, Nguyen MC, Pham TH et al. A new text semi-supervised multi-label learning model based on using the label-feature relations. In: Nguyen N, Pimenidis E, Khan Z, Trawiski B (editors). Computational Collective Intelligence. ICCCI 2018. Lecture Notes in Computer Science. Bristol, UK: Springer, 2018; 11055: pp. 403-413.

[8] Peikari M, Salama S, Nofech-Mozes S, Martel AL. A cluster-then-label semi-supervised learning approach for pathology image classification. Scientific Reports 2018; 8 (1): 7193. doi: 10.1038/s41598-018-24876-0

[9] Raziab S, Karami Mollaei MR, Ghasemib MJ. A novel method for classification of BCI multi-class motor imagery task based on Dempster Shafer theory. Information Sciences 2019; 484: 14-26.

[10] Xua Z, Lv T, Liu L, Zhang Z, Tan J. A regression-type support vector machine for k-class problem. Neuro Computing 2019; 34: 1-7.

[11] Han W, Xue J, Yan H. Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine. IET Information Security 2019; 13 (2): 109-116.

[12] Arshad A, Riaz J, Jiao L. Semi-supervised deep fuzzy c-mean clustering for imbalanced multi-class classification. IEEE Access 2019; 7: 28100-28112.

[13] Wang C, Xu Z, Wang S, Zhang H. Semi-supervised classification framework of hyperspectral images based on the fusion evidence entropy. Multimedia Tools and Applications 2018; 77 (9): 10615-10633.

[14] Wu X, Zhao L, Akoglu L. A quest for structure: jointly learning the graph structure and semi-supervised classification. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; Turin, Italy; 2018, pp. 87-96.

[15] Zadeh LA. Fuzzy sets. Information and Control 1965; 8 (3): 338-353.

[16] Rtahna S, Karthikeyan T. Fuzzy membership functions in privacy preserving data mining. International Journal of Current Research 2016; 8 (3): 28325-28329.

[17] Gasimov RN, Öztürk G. Separation via polyhedral conic functions. Optimization Methods and Software 2006; 21 (4): 527-540.

[18] Uylas N. Methods based on mathematical optimization for data classification. PhD, Ege University, İzmir, Turkey, 2013.

[19] Uylas Sati N. A binary classification algorithm based on polyhedral conic functions. Düzce University Journal of Science and Technology 2015; 3: 152-161.

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

[21] Uylas Sati N, Ordin B. Application of polyhedral conic functions method in the text classification and comparative analysis. Scientific Programming 2018; 2018: 1-11. doi: 10.1155/2018/5349284

[22] Dua D, Graff C. UCI Machine Learning Repository 2019. Irvine, CA, USA: University of California, School of Information and Computer Science.

[23] Uylas Sati N. A collective learning approach for semi-supervised data classification. Pamukkale University of Journal of Engineering Sciences 2018; 24 (5): 864-869.

[24] Driessens K, Reuteman P, Pfahringer B, Leschi C. Using weighted nearest neighbor to benefit from unlabeled data. In: Proceedings of 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining; Macau, China; 2006. pp. 60-69.

[25] Zhou D, Bousquet O, Lal NT, Westor J, Schölkopf B. Learning with local and global consistency. Advances in Neural Information Processing Systems 2004; 321-328.

[26] Garner SR. Weka: the waikato environment for knowledge analysis. In: Proceedings of New Zealand Computer Science Research Students Conference 1995; University of Waikato, Hamilton, New Zealand; 1995. pp. 57-64.

[27] Fazakis N, Karlos S, Kotsiantis S, Sgarbas K. Self-trained LMT for semisupervised learning. Computational Intelligence and Neuroscience 2016; 2016, Article ID 3057481, pp. 1-13.

[28] Schwalbe U. Algorithms, machine learning, and collusion. Journal of Competition Law & Economics 2018; 14 (4): 568–607.

[29] Iggane M, Ennaji A, Mammass D, Yassa M. Self training using a k-nearest neighbor as a base classifier reinforced by support vector machines. International Journal of Computer Applications 2012; 56 (6): 43-46.

[30] Hady M, Schwenker F. Co-training by committee: a new semi-supervised learning framework. In: Proceedings of the IEEE International Conference on Data Mining Workshops; Pisa, Italy; 2008. pp. 563–572.

[31] Chapelle O, Schölkopf B, Zien A. Semi-Supervised Learning. Cambridge, MA, USA: MIT Press, 2006.

[32] Zhou Y, Goldman S. Democratic co-learning. In: Proceedings of 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI ‘04); Boca Raton, FL, USA; 2004 pp. 594-602.

[33] Zhou ZH, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge Data Engineering 2005; 17 (11): 1529-1541.

[34] Li M., Zhou ZH. SETRED: Self-training with editing. In: Ho TB, Cheung D, Liu H (editors). Advances in Knowledge Discovery and Data Mining. PAKDD 2005. Lecture Notes in Computer Science, vol 3518. Berlin, Germany: Springer, 2005, pp. 611-621.