



ELSEVIER

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Least squares projection twin support vector clustering (LSPTSVC)



B. Richhariya, M. Tanveer*, for the Alzheimer's Disease Neuroimaging Initiative¹

Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India

ARTICLE INFO

Article history:

Received 23 May 2019
 Received in revised form 22 April 2020
 Accepted 1 May 2020
 Available online 5 May 2020

Keywords:

Clustering
 Projection twin support vector machine
 Unsupervised learning
 Alzheimer's disease

ABSTRACT

Clustering is a prominent unsupervised learning technique. In the literature, many plane based clustering algorithms are proposed, such as the twin support vector clustering (TWSVC) algorithm. In this work, we propose an alternative algorithm based on projection axes termed as least squares projection twin support vector clustering (LSPTSVC). The proposed LSPTSVC finds projection axis for every cluster in a manner that minimizes the within class scatter, and keeps the clusters of other classes far away. To solve the optimization problem, the concave-convex procedure (CCCP) is utilized in the proposed method. Moreover, the solution of proposed LSPTSVC involves a set of linear equations leading to very less training time. To verify the performance of the proposed algorithm, several experiments are performed on synthetic and real world benchmark datasets. Experimental results and statistical analysis show that the proposed LSPTSVC performs better than existing algorithms w.r.t. clustering accuracy as well as training time. Moreover, a comparison of the proposed method with existing algorithms is presented on biometric and biomedical applications. Better generalization performance is achieved by proposed LSPTSVC on clustering of facial images, and Alzheimer's disease data.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

Twin support vector machine (TWSVM) [15] is one of the most efficient techniques for classification [35,25] and regression problems [24,42]. Applications of TWSVM range from protein localization [44] to classification of breast cancer [20], EEG signals [31], and Alzheimer's disease [32]. The formulation of TWSVM consists of two constrained optimization problems. These kind of constrained optimization problems have been used in many applications such as non-linear systems with full state constraints [26]. Here, a radial basis function neural network is used to approximate the non-linear function. Also, fuzzy based adaptive control methods are proposed for non-linear systems with full state constraints [37,27]. In the recent times, TWSVM has become a major area of research for classification problems [43]. In TWSVM, twin hyperplanes are constructed for binary classification by solving a pair of quadratic programming problems (QPPs). Many novel models based on TWSVM are proposed by researchers to improve its performance such as fuzzy based models [4,16,28], and with different loss

* Corresponding author.

E-mail addresses: phd1701241001@iiti.ac.in (B. Richhariya), mtanveer@iiti.ac.in (M. Tanveer).

¹ Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

functions [41]. Some SVM based models are also proposed to deal with tensor data [10]. However, least squares twin support vector machine (LSTSVM) [17] is a significant improvement on TWSVM in terms of computation cost. LSTSVM solves a system of linear equations rather than QPPs as in TWSVM. A robust energy-based least squares TWSVM (RELS-TSVM) has been proposed by Tanveer et al. [39]. RELS-TSVM [39] incorporates the structural risk minimization principle, and solves a system of linear equations. A recent comprehensive evaluation of 8 variants of TWSVM based classifiers [38] along with 179 classifiers is reported on 90 UCI benchmark datasets. For the 17 different families of datasets, RELS-TSVM emerged as the top-ranked classifier. A novel approach based on projection axes rather than classifying hyperplanes is proposed as projection twin support vector machine [5], and extended for regression problems in [24,50]. To improve the computation cost, Shao et al. [34] proposed an efficient least squares projection twin support vector machine (LSPTSVM). LSPTSVM has been used for various classification problems [48,22]. However, all the above mentioned algorithms are supervised learning algorithms, needing information about true labels in the training process.

For unsupervised learning i.e., where the labels of training data are unknown, algorithms like k-means clustering [14], and fuzzy c-means (FCM) [2] clustering are proposed in the past. In FCM, clustering is performed based on distance from cluster centres with fuzzy membership value for each cluster. However, plane based clustering algorithms are also proposed, such as the k-plane clustering (kPC) algorithm [3]. In kPC, a plane is constructed for each cluster by solving an eigenvalue problem. Some other plane based clustering algorithms are proposed in [33,28]. In 2015, Wang et al. [45] proposed an unsupervised algorithm termed as twin support vector clustering (TWSVC), improving the proximal plane clustering algorithm [33]. To include regularization in TWSVC, a twin bounded support vector clustering (TBSVC) is proposed [1], leading to improved generalization performance. In order to reduce the computation cost of TWSVC, least squares twin support vector clustering (LSTWSVC) is formulated in [16]. In LSTWSVC, a set of linear equations is solved instead of QPPs. A fuzzy least squares twin support vector clustering (FLTWSVC) [16] is also proposed by including fuzzy membership values for the data points.

In contrast to plane based techniques, projection based algorithms minimize the variance of projected data [50] from centre points of the different classes. Motivated by the work on LSPTSVM, we propose a novel projection based clustering algorithm termed as least squares projection twin support vector clustering (LSPTSVC). The idea of LSPTSVC is to minimize the scatter of the cluster from its centre, while keeping rest of the data points far away on both sides of the cluster as shown in Fig. 1. Moreover, the solution of LSPTSVC is obtained by solving a set of linear equations, leading to lesser computation time. To solve the optimization problem of LSPTSVC, we use the concave-convex procedure (CCCP) [49]. There is no need of any optimization toolbox for the solution of proposed LSPTSVC. For computational efficiency, we use only one projection axis [48] for each cluster in LSPTSVC. The main contributions of this work are as follows:

- A novel projection based clustering algorithm is proposed as an alternative to plane based clustering algorithms.
- In contrast to plane based clustering algorithms like TWSVM and LSTWSVM, proposed LSPTSVC constructs a set of projection axes for the clusters.
- Formulation of the proposed LSPTSVC is presented for linear and non-linear transformation of data.
- Theoretical analysis is presented on the initialization and convergence of LSPTSVC.
- Performance comparison of the proposed LSPTSVC is presented on synthetic and real world benchmark datasets.
- Computational efficiency of the proposed LSPTSVC is demonstrated on large scale datasets.
- To justify the applicability of the proposed algorithm on real world applications, proposed LSPTSVC is applied on biometric and biomedical datasets. LSPTSVC is tested on image data for clustering of faces, and facial expressions. Moreover, LSPTSVC is applied for clustering of MRI (magnetic resonance imaging) data of Alzheimer's disease.

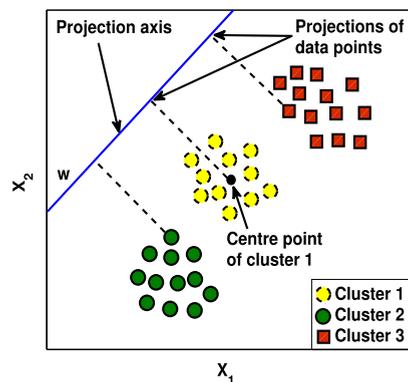


Fig. 1. Clustering by proposed LSPTSVC.

The rest of the paper is organized as follows: Section 2 gives a brief overview of the related work, while Section 3 presents the formulation and analysis of the proposed LSPTSVC. The experimental results are shown in Section 4. Lastly, Section 5 gives the conclusions of the paper with future directions.

The mathematical notations used in this work are as follows: All vectors are assumed as column vectors. X_i is a matrix containing data points belonging to cluster ‘i’ of size $m_i \times n$ and, \bar{X}_i is a matrix containing rest of the data points of size $\bar{m}_i \times n$. Total number of data points are represented by $m = \sum_{i=1}^N m_i$, where N is total number of clusters. The 2-norm of a vector x and matrix X is represented by $\|x\|$ and $\|X\|$ respectively.

2. Related work

In this section, we briefly discuss the formulation of a classification algorithm i.e., LSPTSVM [34], and a clustering algorithm i.e., TWSVC [45].

2.1. Least squares projection twin support vector machine (LSPTSVM)

Linear LSPTSVM [34] generates two non-parallel hyperplanes based on the following optimization problems:

$$\begin{aligned} \min_{w_1} \quad & \frac{1}{2} w_1^T S_1 w_1 + \frac{c_1}{2} \sum_{q=1}^{m_2} (\zeta_q)^2 + \frac{c_3}{2} \|w_1\|^2 \\ \text{s.t.} \quad & w_1^T x_q^{(2)} - w_1^T \frac{1}{m_1} \sum_{p=1}^{m_1} x_p^{(1)} + \zeta_q = 1, \quad q = 0, 1, \dots, m_2, \end{aligned} \tag{1}$$

$$\begin{aligned} \min_{w_2} \quad & \frac{1}{2} w_2^T S_2 w_2 + \frac{c_2}{2} \sum_{p=1}^{m_1} (\eta_p)^2 + \frac{c_4}{2} \|w_2\|^2 \\ \text{s.t.} \quad & - \left(w_2^T x_p^{(1)} - w_2^T \frac{1}{m_2} \sum_{q=1}^{m_2} x_q^{(2)} \right) + \eta_p = 1, \quad p = 0, 1, \dots, m_1, \end{aligned} \tag{2}$$

where $c_i, i = 1, \dots, 4$ are positive parameters, and ζ, η are slack variables. The matrices S_1 and S_2 are written as

$$S_1 = \sum_{p=1}^{m_1} \left(x_p^{(1)} - \frac{1}{m_1} \sum_{p=1}^{m_1} x_p^{(1)} \right) \left(x_p^{(1)} - \frac{1}{m_1} \sum_{p=1}^{m_1} x_p^{(1)} \right)^T, \tag{3}$$

$$S_2 = \sum_{q=1}^{m_2} \left(x_q^{(2)} - \frac{1}{m_2} \sum_{q=1}^{m_2} x_q^{(2)} \right) \left(x_q^{(2)} - \frac{1}{m_2} \sum_{q=1}^{m_2} x_q^{(2)} \right)^T. \tag{4}$$

Now, QPP (1) can be written using matrices of data points in the objective function [34],

$$\begin{aligned} L = \frac{1}{2} w_1^T S_1 w_1 + \frac{c_1}{2} \| -X_2 w_1 + \frac{1}{m_1} e_2 e_1^T X_1 w_1 + e_2 \|^2 \\ + \frac{c_3}{2} \|w_1\|^2, \end{aligned} \tag{5}$$

where e_1, e_2 are vectors of ones of appropriate dimensions.

Setting the gradient of Eq. (5) w.r.t. w_1 equal to 0 and solving, we get

$$\begin{aligned} w_1 = \left(\frac{S_1}{c_1} + \left(-X_2 + \frac{1}{m_1} e_2 e_1^T X_1 \right)^T \left(-X_2 + \frac{1}{m_1} e_2 e_1^T X_1 \right) + \frac{c_3}{c_1} I \right)^{-1} \\ \left(X_2 - \frac{1}{m_1} e_2 e_1^T X_1 \right)^T e_2, \end{aligned} \tag{6}$$

where I is identity matrix of appropriate dimension.

Similarly, w_2 is calculated as

$$w_2 = - \left(\frac{S_2}{c_2} + \left(X_1 - \frac{1}{m_2} e_1 e_2^T X_2 \right)^T \left(X_1 - \frac{1}{m_2} e_1 e_2^T X_2 \right) + \frac{c_4}{c_2} I \right)^{-1} \left(X_1 - \frac{1}{m_2} e_1 e_2^T X_2 \right)^T e_1. \quad (7)$$

For a testing sample x_t , the class is determined as follows,

$$\text{class}(x_t) = \underset{i=1,2}{\operatorname{argmin}} \left| w_i^T x_t - w_i^T \frac{1}{m_i} \sum_{k=1}^{m_i} x_k^{(i)} \right|. \quad (8)$$

2.2. Twin support vector clustering (TWSVC)

TWSVC [45] generates twin non-parallel clustering hyperplanes by solving the following optimization problem.

$$\begin{aligned} \min_{w_i^{j+1}, b_i^{j+1}, \zeta_i^{j+1}} \quad & \frac{1}{2} \| \bar{X}_i w_i^{j+1} + b_i^{j+1} e \|^2 + c_1 e^T \zeta_i^{j+1} \\ \text{s.t.} \quad & T \left(\bar{X}_i w_i^{j+1} + b_i^{j+1} e \right) \geq e - \zeta_i^{j+1}, \zeta_i^{j+1} \geq 0, \\ & i = 0, 1, \dots, N, \end{aligned} \quad (9)$$

where $c_1 > 0$ is the penalty parameter, $T(\cdot)$ is the Taylor series expansion, and ζ_i^{j+1} is the slack variable, $j = 0, 1, \dots$, and e is vector of ones of appropriate dimension.

By using the subgradient [6] of $| \bar{X}_i w_i^j + b_i^j e |$ w.r.t. w_i^j and b_i^j and the Taylor series expansion [45,16], we get

$$\begin{aligned} \min_{w_i^{j+1}, b_i^{j+1}, \zeta_i^{j+1}} \quad & \frac{1}{2} \| \bar{X}_i w_i^{j+1} + b_i^{j+1} e \|^2 + c_1 e^T \zeta_i^{j+1} \\ \text{s.t.} \quad & \operatorname{diag} \left(\operatorname{sign} \left(\bar{X}_i w_i^j + b_i^j e \right) \right) \left(\bar{X}_i w_i^{j+1} + b_i^{j+1} e \right) \geq e - \zeta_i^{j+1}, \zeta_i^{j+1} \geq 0. \end{aligned} \quad (10)$$

The dual problem of QPP (9) is written as

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2} \lambda^T B \left(A^T A \right)^{-1} B^T \lambda - e^T \lambda \\ \text{s.t.} \quad & 0 \leq \lambda \leq c_1 e, \end{aligned} \quad (11)$$

where $B = \operatorname{diag} \left(\operatorname{sign} \left(\bar{X}_i w_i^j + b_i^j e \right) \right) [\bar{X}_i \ e]$, $A = [X_i \ e]$, and λ is the vector of Lagrange multipliers.

The hyperplane for each cluster is found using the following equation,

$$\left[w_i^{j+1} \ b_i^{j+1} \right]^T = \left(A^T A \right)^{-1} B^T \lambda, \quad i = 0, 1, \dots, N. \quad (12)$$

3. Proposed algorithm

In this section, we present the formulations of proposed least squares projection twin support vector clustering (LSPTSVC) for linear and non-linear case. We further discuss the initialization and convergence of LSPTSVC.

The idea of proposed scheme is illustrated in Fig. 1, where a projection axis is generated to cluster the data points. Proposed LSPTSVC minimizes the scatter of a cluster, while keeping the data points of other clusters far away. We also include the regularization term in the objective function to control the structural risk of the model [7]. The regularization term also helps in avoiding the ill-conditioning of the matrices for calculating the inverse [1].

3.1. Linear LSPTSVC

The optimization problem of linear LSPTSVC is described as

$$\begin{aligned}
 \min_{w_i^{j+1}} & \frac{1}{2} \sum_{p=1}^{m_i} \left((w_i^{j+1})^T x_p^{(i)} - (w_i^{j+1})^T \frac{1}{m_i} \sum_{p=1}^{m_i} x_p^{(i)} \right)^2 + \frac{c_1}{2} \sum_{q=1}^{\bar{m}_i} (\zeta_{iq}^{j+1})^2 + \frac{c_2}{2} \|w_i^{j+1}\|^2 \\
 \text{s.t.} & \left| (w_i^{j+1})^T \bar{x}_q^{(i)} - (w_i^{j+1})^T \frac{1}{m_i} \sum_{p=1}^{m_i} x_p^{(i)} \right| + \zeta_{iq}^{j+1} = 1, \\
 & q = 0, 1, \dots, \bar{m}_i, \quad i = 0, 1, \dots, N,
 \end{aligned} \tag{13}$$

where $c_1, c_2 > 0$ are parameters, w_i^{j+1} represents the weight vector of $(j + 1)^{th}$ iteration, $j = 0, 1, \dots$, and the slack variable is represented by ζ_{iq}^{j+1} . The data points of a cluster, and rest of the clusters are represented by $x_p^{(i)}$ and $\bar{x}_q^{(i)}$ respectively. Here, N is the number of clusters, and $\bar{m}_i = (m - m_i)$.

The QPP (13) is formulated by setting an objective function that minimizes intra class variance, while maximizing the inter class distance using the constraints. To solve this optimization problem, we use the concave-convex (CCCP) procedure [49]. Thus, the objective function of QPP (13) can be rewritten as

$$\begin{aligned}
 \min_{w_i^{j+1}} & \frac{1}{2} (w_i^{j+1})^T S_i w_i^{j+1} + \frac{c_1}{2} \sum_{q=1}^{\bar{m}_i} (\zeta_{iq}^{j+1})^2 + \frac{c_2}{2} \|w_i^{j+1}\|^2 \\
 \text{s.t.} & T \left(\left| \bar{X}_i w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^{j+1} \right| \right) + \zeta_{iq}^{j+1} = \bar{e}_i,
 \end{aligned} \tag{14}$$

where $T(\cdot)$ is the first order Taylor series expansion, e_i and \bar{e}_i represent the vector of ones of size p and q respectively. The matrix S_i is written as

$$S_i = \sum_{p=1}^{m_i} (x_p^{(i)} - s_i) (x_p^{(i)} - s_i)^T, \tag{15}$$

where $s_i = \frac{1}{m_i} \sum_{p=1}^{m_i} x_p^{(i)}$ is the centre point of each cluster. Eq. (15) can be rewritten as

$$S_i = (X_i - e_i s_i^T) (X_i - e_i s_i^T)^T. \tag{16}$$

The QPP (14) can be rewritten by substituting the constraints in the objective function as

$$\begin{aligned}
 L = & \frac{1}{2} (w_i^{j+1})^T S_i w_i^{j+1} + \frac{c_1}{2} \| \cdot \| - T \left(\left| \bar{X}_i w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^{j+1} \right| \right) + \bar{e}_i \|^2 \\
 & + \frac{c_2}{2} \|w_i^{j+1}\|^2.
 \end{aligned} \tag{17}$$

Now, the value of the Taylor series expansion is written by using the subgradient [6] of $\left| \bar{X}_i w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^j \right|$ w.r.t. w_i^j [16] as

$$\begin{aligned}
 T \left(\left| \bar{X}_i w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^{j+1} \right| \right) = & \text{diag} \left(\text{sign} \left(\bar{X}_i w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^j \right) \right) \\
 & \left(\bar{X}_i w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^{j+1} \right).
 \end{aligned} \tag{18}$$

Substituting the value of $T(\cdot)$ in Eq. (17), we get

$$\begin{aligned}
 L = & \frac{1}{2} (w_i^{j+1})^T S_i w_i^{j+1} + \frac{c_2}{2} \|w_i^{j+1}\|^2 + \\
 & \frac{c_1}{2} \left\| - \text{diag} \left(\text{sign} \left(\bar{X}_i w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^j \right) \right) \left(\bar{X}_i w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^{j+1} \right) + \bar{e}_i \right\|^2.
 \end{aligned} \tag{19}$$

Solving the gradient of Eq. (19) w.r.t. w_i^{j+1} and equating to 0, we get

$$\begin{aligned}
 S_i w_i^{j+1} + c_2 w_i^{j+1} + c_1 G_i^T \left(G_i w_i^{j+1} - \bar{e}_i \right) = & 0, \\
 \text{where } G_i = & \text{diag} \left(\text{sign} \left(\bar{X}_i w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T X_i w_i^j \right) \right) \left(\bar{X}_i - \frac{1}{m_i} \bar{e}_i e_i^T X_i \right).
 \end{aligned} \tag{20}$$

Solving Eq. (20) for w_i^{j+1} , we get

$$w_i^{j+1} = \left(G_i^T G_i + \frac{S_i}{c_1} + \frac{c_2}{c_1} I_i \right)^{-1} G_i^T \bar{e}_i. \quad (21)$$

For a testing sample x_t , the label y is determined by the following formula,

$$y(x_t) = \arg \min_{i=1,2,\dots,N} \left| w_i^T x_t - \frac{1}{m_i} e_i^T X_i w_i \right|. \quad (22)$$

For the initialization of the labels, nearest neighbour graph (NNG) [45] algorithm is used in LSPTSVC. The algorithm for linear LSPTSVC is shown in Alg. 1.

Algorithm 1 Linear LSPTSVC

1: Inputs :

1.1 Unlabelled data $X \in \mathbb{R}^p$.

2: Initialization :

2.1 Label assignment: $Y_0 \leftarrow \text{NNG}(X)$.

2.2 Initialize weight vector w_i^0 for each cluster $i = 0, 1, \dots, N$:

$w_i^0 = \text{Eigenvector}(S_i)$, for smallest eigenvalue of S_i .

3: CCCP process :

3.1 For each cluster i , calculate w_i^{j+1} for $j = 0$ using the initial weight vector w_i^0 .

3.1.1 $w_i^{j+1} = \text{LSPTSVC}(X, w_i^j)$ using Eq. (21).

3.1.2 **if** ($\|w_i^{j+1} - w_i^j\| > \text{tolerance}$)

$j = j + 1$

go to step 3.1.1

3.1.3 **else**

go to step 4

4: Assign cluster labels :

4.1 Assign new labels to the data points for each cluster i .

4.1.1 $Y_{k+1} = \text{Decision_function}(X, w_i)$ using Eq. (22), initially with $k = 0$.

4.1.2 **if** ($\|Y_{k+1} - Y_k\| \neq 0$)

$k = k + 1$

go to step 3

4.1.3 **else**

go to step 5

5: Output :

Return data labels Y , and projection vectors $w_i, i = 0, 1, \dots, N$.

3.2. Non-linear LSPTSVC

The optimization problem of non-linear LSPTSVC is described as

$$\begin{aligned} \min_{w_i^{j+1}} & \frac{1}{2} \left(w_i^{j+1} \right)^T Z_i w_i^{j+1} + \frac{c_1}{2} \sum_{q=1}^{\bar{m}_i} \left(\xi_{iq}^{j+1} \right)^2 + \frac{c_2}{2} \|w_i^{j+1}\|^2 \\ \text{s.t. } & T \left(\left| K(\bar{X}_i, M^T) w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) w_i^{j+1} \right| \right) + \xi_{iq}^{j+1} = \bar{e}_i, \\ & q = 0, 1, \dots, \bar{m}_i, \quad i = 0, 1, \dots, N, \end{aligned} \quad (23)$$

where $c_1, c_2 > 0$ are parameters, $j = 0, 1, \dots$, and ξ_{iq}^{j+1} represents the slack variable, $T(\cdot)$ is the first order Taylor series expansion, $K(\cdot, M^T)$ is the kernel function [48], and $M = [X_1, X_2, \dots, X_N]$. The matrix Z_i is written as

$$Z_i = \sum_{p=1}^{m_i} \left(K(x_p^{(i)}, M^T) - z_i \right) \left(K(x_p^{(i)}, M^T) - z_i \right)^T, \quad (24)$$

where $z_i = \frac{1}{m_i} \sum_{p=1}^{m_i} K(x_p^{(i)}, M^T)$. Now, Eq. (24) can be rewritten as

$$Z_i = \left(K(X_i, M^T) - e_i z_i^T \right)^T \left(K(X_i, M^T) - e_i z_i^T \right). \tag{25}$$

Now, QPP (23) can be written by using the constraints in the objective function as

$$L = \frac{1}{2} \left(w_i^{j+1} \right)^T Z_i w_i^{j+1} + \frac{c_1}{2} \| \cdot \| - T \left(\left| K(\bar{X}_i, M^T) w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) w_i^{j+1} \right| \right) + \bar{e}_i \|^2 + \frac{c_2}{2} \| w_i^{j+1} \|^2. \tag{26}$$

Substituting the value of T(.) in (26) for the CCCP procedure, we get

$$L = \frac{1}{2} \left(w_i^{j+1} \right)^T Z_i w_i^{j+1} + \frac{c_2}{2} \| w_i^{j+1} \|^2 + \frac{c_1}{2} \| \cdot \| - \text{diag} \left(\text{sign} \left(K(\bar{X}_i, M^T) w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) w_i^j \right) \right) \left(K(\bar{X}_i, M^T) w_i^{j+1} - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) w_i^{j+1} \right) + \bar{e}_i \|^2. \tag{27}$$

Now, solving the gradient of (27) w.r.t. w_i^{j+1} and equating to 0, we get

$$Z_i w_i^{j+1} + \frac{c_2}{2} w_i^{j+1} + c_1 U^T \left(U w_i^{j+1} - \bar{e}_i \right) = 0, \tag{28}$$

where $U_i = \text{diag} \left(\text{sign} \left(K(\bar{X}_i, M^T) w_i^j - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) w_i^j \right) \right) \left(K(\bar{X}_i, M^T) - \frac{1}{m_i} \bar{e}_i e_i^T K(X_i, M^T) \right),$

Solving Eq. (28) for w_i^{j+1} , we get

$$w_i^{j+1} = \left(U_i^T U_i + \frac{Z_i}{c_1} + \frac{c_2}{c_1} I_i \right)^{-1} U_i^T \bar{e}_i. \tag{29}$$

The above equation involves matrix inversion of order $m \times m$. This leads to high computation time for datasets having very large m as compared to number of features. In order to reduce the computation cost of calculating the inverse, we use the Sherman–Morrison–Woodbury (SMW) formula [11]. We can write Eq. (29) as

$$w_i^{j+1} = \left(U_i^T U_i + \frac{c_2}{c_1} I_i + \frac{D_i^T D_i}{c_1} \right)^{-1} U_i^T \bar{e}_i, \tag{30}$$

where $D = \left(K(X_i, M^T) - e_i z_i^T \right)$. Now, using the SMW formula in (30), we obtain the following expression

$$w_i^{j+1} = \left(A_i^{-1} - A_i^{-1} U_i^T \left(I_i + U_i A_i^{-1} U_i^T \right)^{-1} U_i A_i^{-1} \right) U_i^T \bar{e}_i, \tag{31}$$

where $A_i^{-1} = \frac{c_1}{c_2} \left(I_i - D_i^T \left(c_2 I_i + D_i D_i^T \right)^{-1} D_i \right)$. For calculating w_i^{j+1} , instead of computing inverse of size $(m \times m)$, we need to compute one inverse of size $(m_i \times m_i)$, and other of size $(m - m_i) \times (m - m_i), \forall i = 1, 2, \dots, N$.

For a testing sample x_t , the label y is determined as follows,

$$y(x_t) = \arg \min_{i=1,2,\dots,N} \left| w_i^T K(x_t, M^T) - \frac{1}{m_i} e_i^T K(X_i, M^T) w_i \right|. \tag{32}$$

The algorithm for non-linear LSPTSVC is shown in Alg. 2.

Algorithm 2 Non-linear LSPTSVC

1: Inputs :

- 1.1 Unlabelled data $X \in \mathbb{R}^n$.
- 1.2 Kernel matrix K obtained using kernel function.

2: Initialization :

- 2.1 Label assignment: $Y_0 \leftarrow NNG(K)$.

(continued on next page)

* (continued)

Algorithm 2 Non-linear LSPTSVC

2.2 Initialize weight vector w_i^0 for each cluster $i = 0, 1, \dots, N$:
 $w_i^0 = \text{Eigenvector}(Z_i)$, for smallest eigenvalue of Z_i .

3: CCCP process :

3.1 For each cluster i , calculate w_i^{j+1} for $j = 0$ using the initial weight vector w_i^0 .

3.1.1 $w_i^{j+1} = \text{LSPTSVC}(K, w_i^j)$ using Eq. (31).

3.1.2 **if** ($\|w_i^{j+1} - w_i^j\| > \text{tolerance}$)

$j = j + 1$

go to step 3.1.1

3.1.3 **else**

go to step 4

4: Assign cluster labels :

4.1 Assign new labels to the data points for each cluster i .

4.1.1 $Y_{k+1} = \text{Decision_function}(K, w_i)$ using Eq. (32), initially with $k = 0$.

4.1.2 **if** ($\|Y_{k+1} - Y_k\| \neq 0$)

$k = k + 1$

go to step 3

4.1.3 **else**

go to step 5

5: Output :

Return data labels Y , and projection vectors $w_i, i = 0, 1, \dots, N$.

Lemma 3.1. Let $X \in \mathbb{R}^{m \times n}, S \in \mathbb{R}^{n \times n}, m > n$. Then, $S = (X - es)^T(X - es)$ is a positive semidefinite matrix, where $s = \frac{1}{m} \sum_{p=1}^m x_p^T$.

Proof: Let $T = (X - es)$. Then, $S = T^T T$ is a symmetric matrix.
 Now, for any $w \in \mathbb{R}^n$,

$$\begin{aligned} w^T S w &= w^T T^T T w \\ \Rightarrow w^T S w &= \|T w\|^2 \geq 0. \end{aligned} \quad (33)$$

Therefore, S is positive semidefinite.

Theorem 3.1. Let $X \in \mathbb{R}^{p \times n}, m > n$, and $S = (X - es)^T(X - es)$. Then, the global minimum of:

$$\begin{aligned} \min_w \quad & w^T S w \\ \text{s.t.} \quad & w^T w = 1, \end{aligned} \quad (34)$$

is obtained for any eigenvector w of S with minimum eigenvalue. The minimum value of (34) is positive iff S is positive definite or equivalently iff $\text{rank}(X - es) = n$.

Proof: Firstly, we write the Lagrangian of Eq. (34),

$$L = w^T S w - \lambda(w^T w - 1). \quad (35)$$

Now, we use the Karush-Kuhn-Tucker (K.K.T.) optimality conditions,

$$\frac{\partial L}{\partial w} = S w - \lambda w = 0, \quad (36)$$

$$\frac{\partial L}{\partial \lambda} = w^T w - 1 = 0. \quad (37)$$

From (36) and (37), we get

$$\lambda = w^T S w. \quad (38)$$

Putting the value of λ in (36), we get

$$S w = (w^T S w) w, \quad (39)$$

which is equivalent to

$$Sw = kw, \tag{40}$$

where $k = w^T Sw$, which is to be minimized in Eq. (34). Hence, the smallest eigenvalue of S gives the eigenvector w to achieve global minimum of (34) [3].

Remark 1. If $S \in \mathbb{R}^{n \times n}$ is positive definite, then S is non-singular.

Theorem 3.2. Let $X \in \mathbb{R}^{p \times n}, G \in \mathbb{R}^{q \times n}, S = (X - es)^T(X - es) \in \mathbb{R}^{n \times n}$, and $I \in \mathbb{R}^{n \times n}$ is the identity matrix. Then, the matrix $(G^T G + \frac{S}{c_1} + \frac{c_2}{c_1} I)$ is invertible $\forall c_1, c_2 > 0$.

Proof: Here, $G^T G$ and S are positive semidefinite matrices from Lemma 3.1, and $\frac{c_2}{c_1} I_i$ is positive definite for $c_1, c_2 > 0$. Also, for any vector $w \in \mathbb{R}^n, w \neq 0$, the sum of a positive semidefinite and positive definite matrix is always positive definite as shown below:

Let A and B be a positive definite and positive semidefinite matrix respectively. Then, for any vector $w \in \mathbb{R}^n, w \neq 0$,

$$w^T A w > 0, \tag{41}$$

$$w^T B w \geq 0, \tag{42}$$

Adding (41) & (42), we get

$$(w^T A w + w^T B w) > 0, \tag{43}$$

$$w^T (A + B) w > 0. \tag{44}$$

Therefore, the square matrix $(G^T G + \frac{S}{c_1} + \frac{c_2}{c_1} I)$ is always non-singular (Remark 1), and thus invertible.

3.3. Convergence

Proposed LSPTSVC described in Alg. 1 and 2 converges in a finite number of steps. This is because CCCP method always finds a local minimum, and thus converges as discussed in [49]. Moreover, in the cluster assignment process, every data point is assigned to closest hyperplane [3]. So, the overall objective function cannot increase. Thus, the algorithm converges based on any of the following terminating conditions:

- (i) Same labels assigned to data points in two consecutive iterations.
- (ii) Non-decrease in the overall objective function.

3.4. Time complexity

In comparison to TWSVC which solves large sized QPPs to solve the clustering problem, proposed LSPTSVC only needs to solve sets of linear equations. The time complexity of solving the QPP in linear TWSVC is $O(N(\bar{m}_i)^3)$ for $m = m_i + \bar{m}_i$ samples, N classes and \bar{m}_i constraints, $i = 1, 2, \dots, N$. The complexity of calculating the matrix inverse is about $O(Nn^3)$ [1]. So, the complexity of TWSVC becomes $O(N(\bar{m}_i^3 + n^3))$. In case of non-linear TWSVC, the complexity for QPP is $O(N\bar{m}_i^3)$, where \bar{m}_i is the number of constraints, and for inverse is about $O(Nm^3)$. Therefore, the complexity becomes $O(N(\bar{m}_i^3 + m^3))$. The time complexity of TBSVC is same as TWSVC.

The solution of linear LSPTSVC requires the inversion of N matrices of size $n \times n$. Thus, the time complexity of solving the inverses in Eq. (21) is $O(Nn^3)$. In the non-linear case, N matrix inverses of size m_i , and \bar{m}_i need to be calculated. Therefore, the time complexity is $O(N(m_i^3 + \bar{m}_i^3)), i = 1, 2, \dots, N$. The time complexity of LSPTSVC is lower than TWSVC which leads to lesser training time. LSTWSVC has similar time complexity as LSPTSVC, except the fact that it needs to calculate the bias. Consequently, the number of linear equations in LSPTSVC is less than LSTWSVC by one equation.

Moreover, in the initialization process, proposed LSPTSVC only needs to find w . On the other hand, the bias b is also calculated in case of existing plane based clustering algorithms.

3.5. Proposed LSPTSVC vs LSPTSVM

The LSPTSVM [34] is a supervised learning algorithm that performs classification of data points by constructing projection axes for each class. The decision function shown in Eq. (8) is constructed in a manner to keep the projected data well separated. However, LSPTSVM minimizes the within class variance of one class, and keeps the scatter of the other class far away on one side of the axis. This is shown by the constraints of QPP (1).

We extended the projection axes based approach for unsupervised learning. Proposed LSPTSVC performs clustering by minimizing the within cluster variance, and keeping the scatter of the other clusters far away on both sides of the axes as shown in Fig. 1. This is a result of the constraints of QPP (13). Moreover, LSPTSVM solves the optimization problem by system of linear equations, whereas proposed LSPTSVC solves linear equations in multiple iterations of the CCCP procedure to obtain the projection axes.

In contrast to LSPTSVM, proposed LSPTSVC involves an initialization procedure for the weights. Since, the scatter matrix S involved in the projection based algorithms is positive semidefinite, we presented the initialization procedure based on eigenvalue of S . In terms of time complexity, LSPTSVC requires more computation time than LSPTSVM, since it involves the CCCP iterative procedure, and mostly deals with multiclass clustering of data. However, proposed LSPTSVC is computationally more efficient than LSTWSVC. This is analogous to the lesser computation cost of LSPTSVM in comparison to LSTWSVM [34].

4. Experimental results

In this section, performance of the proposed LSPTSVC is compared with existing techniques on the basis of clustering accuracy and training time. The algorithms used for comparison are FCM [2], kPC [3], TWSVC [45], TBSVC [1], LSTWSVC [16], and FLSTWSVC [16]. Among these, FCM is a distance based technique using fuzzy memberships, while rest are plane based algorithms. We use 13 synthetic and 10 real world benchmark datasets to assess the performance of the proposed model with linear and non-linear kernels. Moreover, the performance of LSPTSVC is compared with existing algorithms on 12 large scale datasets. Performance comparison on real world applications are also presented viz. clustering of faces, facial expressions, and Alzheimer's disease data.

4.1. Data

The synthetic benchmark datasets are downloaded from the website (<https://github.com/deric/clustering-benchmark>), while the real world and large scale datasets are taken from UCI repository [9]. For applications, facial images are downloaded from AT&T database (<https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>) of AT&T Laboratories Cambridge, and facial expression data is taken from JAFFE database [21].

All MRI images used in this work were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The main goal of ADNI is to find out the effectiveness of neuroimaging techniques like MRI, positron emission tomography (PET), other biological markers, and clinical neuropsychological tests to estimate the onset of Alzheimer's disease from the state of mild cognitive impairment. For more information, visit www.adni-info.org.

4.2. Experimental setup

All the computations are carried out on a PC running on 64 bit Windows 10 operating system, 3.60 GHz Intel® core™ i7-7700 processor, 16 GB of RAM under MATLAB R2008b environment. MOSEK optimization toolbox (<http://www.mosek.com>) is used to solve the QPPs in case of TWSVC and TBSVC. The parameter selection is performed using 5-fold cross-validation for all the methods. For non-linear case, Gaussian kernel is used in all the methods, defined as

$$K(x, y) = \exp\left(\frac{-1}{2\mu^2} \|x - y\|^2\right), \quad (45)$$

where x and y are vectors and μ is a parameter.

In Tables 1 and 2, 50% of total data samples are used for training and rest for testing. The value of the parameters c_1, c_2 are selected from the set $\{10^{-5}, 10^{-4}, \dots, 10^5\}$, while μ is chosen from the set $\{2^{-5}, 2^{-4}, \dots, 2^5\}$ for all the cases. The tolerance value for the CCCP process is set as 0.001 in all the algorithms. In FCM, the weighting exponent i.e., m is selected from the set $\{1.25, 1.5, 1.75, 2\}$ [2]. In case of large datasets, the value of c_1, c_2 is fixed as 1 [34], and μ is set as 2^5 for all the algorithms. The clustering accuracy for l data samples with y labels is measured using the following similarity matrix $L \in \mathbb{R}^{l \times l}$ [45],

$$L(i, j) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise.} \end{cases}$$

Now, let L_p is similarity matrix of predicted cluster labels, and L_a is the similarity matrix of actual labels. Then, the accuracy is defined as the rand index [45],

$$\text{Accuracy} = \frac{n_0 + n_1 - l}{l^2 - l} \times 100\%, \quad (46)$$

where n_0 is the number of zeros in L_a and L_p , and n_1 is the number of ones in L_a and L_p .

Table 1

Performance comparison on clustering accuracy (%), number of mis-clustered data points (# Miss), and training time of the proposed LSPTSVC with existing algorithms using linear kernel. The Win-Tie-Loss calculation is based on accuracy, and 's' represents time in seconds.

Dataset (Size, clusters)	FCM [2] Accuracy # Miss - Time (s)	kPC [3] Accuracy # Miss - Time (s)	TWSVC [45] Accuracy # Miss (c ₁) Time (s)	TBSVC [1] Accuracy # Miss (c ₁ , c ₂) Time (s)	LSTWSVC [16] Accuracy # Miss (c ₁ , c ₂) Time (s)	FLSTWSVC [16] Accuracy # Miss (c ₁ , c ₂) Time (s)	Proposed LSPTSVC Accuracy # Miss (c ₁ , c ₂) Time (s)
Synthetic							
3MC (400 × 2, 3)	51.37 89 - 0.0151	66.3 89 - 0.00008	68.96 80 (10 ⁰) 0.0191	90.1 18 (10 ⁻⁵ , 10 ²) 0.0162	90.1 18 (10 ⁻⁵ , 10 ²) 0.00015	89.29 16 (10 ¹ , 10 ¹) 0.0037	96.02 6 (10 ² , 10 ⁵) 0.00009
Aggregation (788 × 2, 7)	77.64 152 - 0.1394	79.36 189 - 0.00015	79.97 183 (10 ⁻¹) 0.2811	78.86 178 (10 ⁰ , 10 ⁰) 0.2869	80.82 187 (10 ⁻³ , 10 ⁻¹) 0.00324	89.62 95 (10 ⁻¹ , 10 ⁻¹) 0.0418	83.55 149 (10 ¹ , 10 ⁵) 0.00185
Compound (399 × 2, 6)	78.41 72 - 0.0718	73.56 97 - 0.0001	79.88 81 (10 ⁰) 0.055	80.14 84 (10 ⁻¹ , 10 ⁻⁴) 0.054	79.88 92 (10 ⁻³ , 10 ⁻⁴) 0.00039	79.76 80 (10 ⁻² , 10 ⁻²) 0.0077	86.61 51 (10 ⁻⁵ , 10 ³) 0.00026
R15 (600 × 2, 15)	92.17 126 - 0.0243	91.4 133 - 0.00019	96.18 64 (10 ⁻²) 0.3949	93.49 102 (10 ⁻¹ , 10 ⁻³) 0.4091	95.58 75 (10 ⁻³ , 10 ⁻⁵) 0.00238	97.71 36 (10 ⁰ , 10 ⁻⁵) 0.0486	97.07 48 (10 ⁰ , 10 ⁴) 0.00123
Zelnik5 (512 × 2, 4)	68.73 112 - 0.0163	78.73 96 - 0.00011	78.43 99 (10 ²) 0.055	95.08 15 (10 ¹ , 10 ⁻¹) 0.0472	85.99 58 (10 ⁻⁵ , 10 ⁻²) 0.00044	82.75 72 (10 ⁻⁵ , 10 ⁻⁵) 0.0107	91.4 26 (10 ² , 10 ⁻⁵) 0.00022
2d-4c-no9 (876 × 2, 4)	85.77 114 - 0.0931	77.66 130 - 0.00014	97.49 14 (10 ⁻²) 0.1426	96.85 18 (10 ⁻¹ , 10 ¹) 0.1436	97.01 17 (10 ⁻¹ , 10 ¹) 0.00169	96.37 22 (10 ⁰ , 10 ⁻⁵) 0.0244	98.59 8 (10 ⁻³ , 10 ²) 0.00084
Longsquare (900 × 2, 6)	81.67 179 - 0.0114	83.46 159 - 0.00016	84.27 127 (10 ⁻⁵) 0.228	86.47 128 (10 ⁻⁵ , 10 ⁻¹) 0.2248	85.78 130 (10 ⁻⁴ , 10 ⁻¹) 0.00408	90.15 101 (10 ⁻² , 10 ⁻⁴) 0.0537	90.93 94 (10 ² , 10 ⁴) 0.00259
Hepta (212 × 3, 7)	77.05 43 - 0.006	87.57 38 - 0.00011	99.03 2 (10 ⁻³) 0.0268	99.03 2 (10 ⁻³ , 10 ⁻⁵) 0.0268	98.49 3 (10 ⁻³ , 10 ⁻⁵) 0.00016	94.9 11 (10 ⁻¹ , 10 ⁻²) 0.0031	100 0 (10 ⁻³ , 10 ⁰) 0.00015
Zelnik3 (266 × 2, 3)	74.87 31 - 0.0054	81.6 20 - 0.00007	80.25 22 (10 ⁻⁵) 0.0113	80.25 22 (10 ⁻⁵ , 10 ⁻²) 0.0109	80.25 22 (10 ⁻² , 10 ⁻²) 0.00009	75.1 30 (10 ⁻⁵ , 10 ⁻⁵) 0.0018	83.96 17 (10 ² , 10 ⁻⁵) 0.00006
Pathbased (300 × 2, 3)	61.71 74 - 0.0156	70.36 48 - 0.00007	71.22 45 (10 ⁻¹) 0.0135	71.44 45 (10 ⁻² , 10 ⁻¹) 0.0131	70.7 47 (10 ⁻² , 10 ⁻¹) 0.00011	71.44 45 (10 ⁻⁵ , 10 ⁻⁵) 0.0023	71.35 45 (10 ¹ , 10 ⁵) 0.00008
Zelnik1 (399 × 2, 3)	56.47 70 - 0.022	58.72 71 - 0.00009	56.05 71 (10 ⁻¹) 0.0136	59.59 72 (10 ² , 10 ⁰) 0.0144	56.05 70 (10 ⁻² , 10 ⁻³) 0.00011	49.58 74 (10 ⁰ , 10 ⁻¹) 0.0031	59.65 69 (10 ² , 10 ⁻⁵) 0.00008
Ds2c2sc13 (588 × 2, 13)	87.44 115 - 0.4318	90.46 103 - 0.00035	93.08 69 (10 ⁻³) 0.3133	93.01 70 (10 ⁻³ , 10 ⁻⁵) 0.3154	93.17 72 (10 ⁻³ , 10 ⁻⁵) 0.0021	93.06 79 (10 ⁻¹ , 10 ⁻²) 0.0437	93.45 81 (10 ¹ , 10 ¹) 0.00128
2d-4c-no4 (863 × 2, 4)	87.68 66 - 0.0048	62.52 205 - 0.0003	67.13 131 (10 ²) 0.1697	85.22 70 (10 ⁰ , 10 ⁵) 0.0754	85.22 70 (10 ⁻⁵ , 10 ⁵) 0.00261	62.96 205 (10 ³ , 10 ⁻³) 0.0275	93.38 24 (10 ¹ , 10 ⁵) 0.00148
Real world							
Ecoli (336 × 7, 8)	71.02 71 - 0.0144	37.79 81 - 0.00034	63.47 81 (10 ⁵) 0.1092	63.44 82 (10 ³ , 10 ⁴) 0.0563	61.98 80 (10 ¹ , 10 ⁻⁵) 0.00033	56.19 81 (10 ² , 10 ⁵) 0.0083	68.25 84 (10 ¹ , 10 ¹) 0.00035
Zoo (101 × 16, 7)	54.53 24 - 0.0181	72.73 15 - 0.0023	82.04 14 (10 ⁻³) 0.0309	89.8 11 (10 ⁻¹ , 10 ⁰) 0.0217	88.98 11 (10 ⁻⁵ , 10 ³) 0.00026	90.12 11 (10 ⁻¹ , 10 ¹) 0.0057	88.41 12 (10 ² , 10 ⁻⁵) 0.00022

(continued on next page)

Table 1 (continued)

Dataset (Size, clusters)	FCM [2] Accuracy # Miss - Time (s)	kPC [3] Accuracy # Miss - Time (s)	TWSVC [45] Accuracy # Miss - Time (s)	TBSVC [1] Accuracy # Miss (c_1, c_2) Time (s)	LSTWSVC [16] Accuracy # Miss (c_1, c_2) Time (s)	FLSTWSVC [16] Accuracy # Miss (c_1, c_2) Time (s)	Proposed LSPTSVC Accuracy # Miss (c_1, c_2) Time (s)
Wine (178 × 13, 3)	34.22 43 - 0.0048	56.03 43 - 0.00016	73.88 25 (10^{-2}) 0.0087	71.71 26 ($10^{-5}, 10^0$) 0.009	75.74 21 ($10^{-3}, 10^{-1}$) 0.00011	79.01 17 ($10^1, 10^{-3}$) 0.0026	74.06 23 ($10^{-1}, 10^5$) 0.00009
Iris (150 × 4, 3)	32.68 37 - 0.0059	62.7 29 - 0.00007	91.53 5 (10^{-5}) 0.0083	93.08 4 ($10^{-1}, 10^0$) 0.0083	94.7 3 ($10^{-1}, 10^1$) 0.00007	94.7 3 ($10^{-5}, 10^0$) 0.0009	94.7 3 ($10^{-5}, 10^0$) 0.00006
Seeds (210 × 7, 3)	32.77 52 - 0.0072	76.58 33 - 0.00015	75.66 33 (10^{-2}) 0.0116	75.99 34 ($10^{-5}, 10^{-2}$) 0.0103	75.44 35 ($10^{-5}, 10^{-2}$) 0.00008	75.66 33 ($10^{-1}, 10^{-4}$) 0.0022	74.56 27 ($10^2, 10^2$) 0.00008
Teachingeval (151 × 5, 3)	33.98 36 - 0.0047	42.85 37 - 0.00009	48.32 37 (10^2) 0.0093	52.65 37 ($10^{-1}, 10^0$) 0.0083	51.32 37 ($10^{-2}, 10^2$) 0.0001	49.05 37 ($10^{-5}, 10^1$) 0.0011	56.25 36 ($10^1, 10^{-1}$) 0.00006
Tae (150 × 5, 3)	32.68 37 - 0.0088	45.15 36 - 0.0001	52.61 37 (10^3) 0.0094	52.61 37 ($10^5, 10^{-5}$) 0.0098	49.48 35 ($10^1, 10^3$) 0.00011	32.68 37 ($10^{-5}, 10^{-5}$) 0.0012	56.61 36 ($10^2, 10^4$) 0.00006
Hayes-roth (132 × 4, 3)	62.75 27 - 0.0098	57.67 31 - 0.00007	55.99 32 (10^1) 0.0077	55.34 33 ($10^0, 10^{-4}$) 0.0077	53.38 32 ($10^{-2}, 10^{-2}$) 0.00008	52.73 28 ($10^0, 10^{-5}$) 0.0008	53.94 32 ($10^2, 10^3$) 0.00005
Shuttle (1486 × 9, 5)	63.25 167 - 0.3162	51.55 359 - 0.00235	62.85 251 (10^{-1}) 1.1994	87.77 54 ($10^{-2}, 10^2$) 1.4062	78.54 124 ($10^0, 10^5$) 0.00828	82.12 104 ($10^2, 10^4$) 0.1927	78.54 112 ($10^0, 10^4$) 0.00783
Libras (360 × 90, 15)	70.99 89 - 0.4334	64.14 88 - 0.0491	66.16 89 (10^{-5}) 0.1845	83.27 88 ($10^{-1}, 10^{-1}$) 0.1641	86.72 87 ($10^1, 10^{-4}$) 0.00584	87.65 85 ($10^{-4}, 10^{-3}$) 0.0261	87.81 88 ($10^0, 10^{-4}$) 0.00503
Average rank (Accuracy)	5.8043	5.4783	4.1522	3.1522	3.6304	3.7609	2.0217
Average rank (# Miss)	5.1304	5.3696	4.1957	3.9783	3.6304	3.3261	2.3696
Win-Tie-Loss	21-0-2	21-0-2	21-0-2	17-0-6	18-2-3	15-1-7	

For application on Alzheimer's data, Freesurfer's recon-all pipeline (version 6.0.1) [29] is used for processing the MRI images. The volumetric features of the brain are normalized by the respective total intracranial volume (TIV) of the subjects [47,36].

In all the existing algorithms except FCM, the initialization of weights is performed using kPC algorithm, while LSPTSVC is initialized using Theorem 3.1. For initialization of cluster labels, the well known nearest neighbour graph (NNG) technique [45] is used for all the algorithms except FLSTWSVC which uses fuzzy NNG [16]. However, in case of large datasets, a set of randomly generated cluster labels are used for initialization of all the algorithms.

4.3. Results on benchmark datasets

The comparison of the proposed LSPTSVC with existing methods viz. FCM, kPC, TWSVC, TBSVC, LSTWSVC, and FLSTWSVC is shown in Table 1 for linear case. One can observe that proposed LSPTSVC is showing better performance w.r.t. clustering accuracy in comparison to existing algorithms. This is also justified by the lowest average rank of LSPTSVC i.e., 2.0217 for all the datasets. Moreover, the training time of LSPTSVC is lesser than TWSVC and TBSVC. This is due to the fact that the proposed LSPTSVC solves a set of linear equations to obtain the projection axis. In contrast, TWSVC and TBSVC solve computationally expensive QPPs. In comparison to LSTWSVC, LSPTSVC is slightly faster in most datasets since it only needs to calculate the weight vector w and not the bias b . For FLSTWSVC, the training time is higher than proposed LSPTSVC due to the overhead of calculation of fuzzy membership. The comparison of computation time of kPC and FCM with proposed LSPTSVC is not justified, since they are not twin SVM based algorithms. However, we have shown the training time of all the algorithms in the tables.

Table 2

Performance comparison on clustering accuracy (%), number of mis-clustered data points (# Miss), and training time of the proposed LSPTSVC with existing algorithms using Gaussian kernel. The Win-Tie-Loss calculation is based on accuracy, and 's' represents time in seconds.

Dataset (Size, clusters)	FCM [2] Accuracy # Miss - Time (s)	kPC [3] Accuracy # Miss - Time (s)	TWSVC [45] Accuracy # Miss (c_1, μ) Time (s)	TBSVC [1] Accuracy # Miss ($c_1 = c_2, \mu$) Time (s)	LSTWSVC [16] Accuracy # Miss (c_1, c_2, μ) Time (s)	FLSTWSVC [16] Accuracy # Miss (c_1, c_2, μ) Time (s)	Proposed LSPTSVC Accuracy # Miss (c_1, c_2, μ) Time (s)
Synthetic							
3MC (400 × 2, 3)	51.37 89 - 0.0151	69.18 63 - 0.0383	66.32 97 (10 ⁰ , 2 ⁵) 0.0977	60.81 93 (10 ¹ , 2 ⁰) 0.0993	88.85 18 (10 ⁻³ , 10 ⁻¹ , 2 ⁵) 0.045	72.27 62 (10 ³ , 10 ⁻⁴ , 2 ⁴) 0.0455	82.31 29 (10 ¹ , 10 ¹ , 2 ⁵) 0.0444
Aggregation (788 × 2, 7)	77.64 152 - 0.1394	92.21 77 - 0.638	92.36 77 (10 ⁻³ , 2 ³) 0.8281	92.85 70 (10 ⁻⁵ , 2 ¹) 0.7539	92.8 71 (10 ⁻⁵ , 10 ⁻² , 2 ¹) 0.29	94.97 42 (10 ⁻¹ , 10 ⁻¹ , 2 ³) 0.2958	92.99 70 (10 ⁻⁵ , 10 ⁻⁵ , 2 ¹) 0.2835
Compound (399 × 2, 6)	78.41 72 - 0.0718	88.57 53 - 0.1013	91.34 44 (10 ⁻⁵ , 2 ⁰) 0.1506	91.37 41 (10 ⁻² , 2 ¹) 0.1482	93.24 38 (10 ⁻³ , 10 ⁻⁴ , 2 ⁰) 0.0557	90.81 43 (10 ⁻³ , 10 ⁻⁵ , 2 ¹) 0.0588	93.05 40 (10 ⁻² , 10 ² , 2 ²) 0.0527
R15 (600 × 2, 15)	92.17 126 - 0.0243	99.43 8 - 0.6568	99.73 4 (10 ⁻⁵ , 2 ⁵) 0.6917	99.8 3 (10 ⁻⁵ , 2 ⁴) 0.7201	99.42 7 (10 ⁻⁵ , 10 ⁻⁴ , 2 ⁻¹) 0.2268	99.43 7 (10 ⁻⁵ , 10 ⁻⁵ , 2 ⁻¹) 0.23	99.57 5 (10 ⁻⁵ , 10 ⁻⁴ , 2 ⁻¹) 0.2197
Zelnik5 (512 × 2, 4)	68.73 112 - 0.0163	75.96 87 - 0.0946	82.07 76 (10 ⁻³ , 2 ⁵) 0.1574	86.85 53 (10 ⁻² , 2 ¹) 0.1356	84.03 80 (10 ⁻¹ , 10 ⁻² , 2 ¹) 0.0837	85.64 67 (10 ⁻⁵ , 10 ⁻⁵ , 2 ²) 0.0876	84.86 61 (10 ² , 10 ⁵ , 2 ⁻³) 0.0794
2d-4c-no9 (876 × 2, 4)	85.77 114 - 0.0931	98.66 8 - 0.3521	98.39 7 (10 ⁻⁵ , 2 ²) 0.6216	97.81 10 (10 ⁻⁵ , 2 ¹) 0.6215	97.07 15 (10 ⁻² , 10 ¹ , 2 ²) 0.2801	99.16 5 (10 ⁻⁵ , 10 ⁻⁴ , 2 ¹) 0.2892	98.28 8 (10 ⁻⁴ , 10 ⁻² , 2 ²) 0.2798
Longsquare (900 × 2, 6)	81.67 179 - 0.0114	89.27 107 - 0.5345	64.72 200 (10 ⁻² , 2 ¹) 1.0026	78.86 192 (10 ⁰ , 2 ²) 0.8856	81.08 198 (10 ⁻⁴ , 10 ² , 2 ²) 0.3905	93.22 65 (10 ⁻³ , 10 ⁻⁴ , 2 ²) 0.3796	93.76 49 (10 ⁰ , 10 ⁴ , 2 ²) 0.3779
Hepta (212 × 3, 7)	77.05 43 - 0.006	100 0 - 0.018	100 0 (10 ⁻⁵ , 2 ¹) 0.0563	100 0 (10 ⁻⁵ , 2 ¹) 0.0553	100 0 (10 ⁻⁵ , 10 ⁻⁵ , 2 ³) 0.0156	100 0 (10 ⁻⁵ , 10 ⁻⁵ , 2 ⁰) 0.0176	100 0 (10 ⁻⁵ , 10 ⁻⁵ , 2 ¹) 0.0144
Zelnik3 (266 × 2, 3)	74.87 31 - 0.0054	83.12 19 - 0.0253	83.39 18 (10 ⁻¹ , 2 ⁵) 0.048	81.7 20 (10 ⁻⁴ , 2 ²) 0.0455	84.51 17 (10 ¹ , 10 ⁴ , 2 ⁻⁴) 0.0206	100 0 (10 ⁻⁵ , 10 ⁻³ , 2 ⁻⁵) 0.0214	100 0 (10 ⁰ , 10 ³ , 2 ⁻⁴) 0.0192
Pathbased (300 × 2, 3)	61.71 74 - 0.0156	63.9 64 - 0.0252	57.49 74 (10 ⁴ , 2 ⁵) 0.0638	68.21 60 (10 ⁰ , 2 ⁵) 0.0571	70.56 46 (10 ⁻⁵ , 10 ⁰ , 2 ⁵) 0.0256	75.7 34 (10 ⁻² , 10 ⁻⁵ , 2 ²) 0.026	82.26 23 (10 ¹ , 10 ³ , 2 ²) 0.0255
Zelnik1 (399 × 2, 3)	56.47 70 - 0.022	52.6 74 - 0.0217	59.82 69 (10 ⁻² , 2 ⁴) 0.0433	58.52 69 (10 ⁻² , 2 ¹) 0.0381	68.57 57 (10 ² , 10 ⁰ , 2 ⁻⁴) 0.0268	73.41 53 (10 ⁵ , 10 ³ , 2 ⁻⁴) 0.0275	74.16 56 (10 ⁴ , 10 ¹ , 2 ²) 0.0248
Ds2c2sc13 (588 × 2, 13)	87.44 115 - 0.4318	84.25 114 - 0.3848	94.28 45 (10 ⁻⁵ , 2 ⁻³) 0.5756	94.78 43 (10 ⁻³ , 2 ⁻⁴) 0.5546	95.14 41 (10 ⁻⁵ , 10 ⁻⁵ , 2 ⁻⁴) 0.1956	93.47 77 (10 ⁻³ , 10 ⁻⁴ , 2 ⁻³) 0.2095	95 41 (10 ⁻⁵ , 10 ⁻⁴ , 2 ⁻⁵) 0.1902
2d-4c-no4 (863 × 2, 4)	87.68 66 - 0.0048	71.55 101 - 0.3495	65.39 155 (10 ⁻⁴ , 2 ¹) 0.5077	64.93 156 (10 ⁰ , 2 ²) 0.4275	62.14 162 (10 ⁻⁵ , 10 ⁵ , 2 ²) 0.2884	89.65 66 (10 ⁻¹ , 10 ³ , 2 ¹) 0.2889	95.27 20 (10 ¹ , 10 ⁵ , 2 ²) 0.2773
Real world							
Ecoli (336 × 7, 8)	71.02 71 - 0.0144	68.31 69 - 0.0496	61.13 84 (10 ⁻⁵ , 2 ³) 0.1331	60.61 83 (10 ⁴ , 2 ³) 0.1175	63.45 74 (10 ¹ , 10 ⁻⁴ , 2 ⁵) 0.0434	55.01 67 (10 ⁻¹ , 10 ⁰ , 2 ⁵) 0.0466	73.28 81 (10 ¹ , 10 ³ , 2 ⁵) 0.0407
Zoo (101 × 16, 7)	54.53 24 - -	81.71 16 - -	86.53 11 (10 ⁻⁵ , 2 ³) -	86.86 14 (10 ⁻² , 2 ²) -	85.88 13 (10 ³ , 10 ⁻⁵ , 2 ³) -	80.33 15 (10 ⁻¹ , 10 ⁻⁵ , 2 ²) -	89.31 9 (10 ¹ , 10 ⁻¹ , 2 ³) -

(continued on next page)

Table 2 (continued)

Dataset (Size, clusters)	FCM [2] Accuracy # Miss - Time (s)	kPC [3] Accuracy # Miss - Time (s)	TWSVC [45] Accuracy # Miss (c_1, μ) Time (s)	TBSVC [1] Accuracy # Miss (c_1, c_2, μ) Time (s)	LSTWSVC [16] Accuracy # Miss (c_1, c_2, μ) Time (s)	FLSTWSVC [16] Accuracy # Miss (c_1, c_2, μ) Time (s)	Proposed LSPTSVC Accuracy # Miss (c_1, c_2, μ) Time (s)
Wine (178 × 13, 3)	0.0181 34.22 43 -	0.0039 65.27 30 -	0.0379 68.69 27 ($10^{-5}, 2^4$)	0.0311 72.32 22 ($10^{-2}, 2^4$)	0.0039 74.26 23 ($10^{-1}, 10^0, 2^5$)	0.0078 59.24 38 ($10^{-5}, 10^{-2}, 2^5$)	0.0035 75.18 21 ($10^{-1}, 10^{-3}, 2^5$)
Iris (150 × 4, 3)	0.0048 32.68 37 -	0.0067 78.31 16 -	0.0262 94.7 3 ($10^{-1}, 2^5$)	0.0258 94.7 3 ($10^{-5}, 2^5$)	0.0096 96.4 2 ($10^{-1}, 10^{-3}, 2^2$)	0.0103 90.05 6 ($10^{-1}, 10^{-2}, 2^0$)	0.009 94.7 3 ($10^{-1}, 10^{-2}, 2^2$)
Seeds (210 × 7, 3)	0.0059 32.77 52 -	0.0051 76.1 33 -	0.019 76.58 33 ($10^{-5}, 2^5$)	0.0197 87.09 12 ($10^{-3}, 2^5$)	0.0063 87.01 12 ($10^{-5}, 10^{-3}, 2^5$)	0.007 73.79 33 ($10^1, 10^3, 2^3$)	0.0062 87.71 11 ($10^{-1}, 10^2, 2^4$)
Teachingeval (151 × 5, 3)	0.0072 33.98 36 -	0.0127 44 35 -	0.0311 40.76 36 ($10^{-3}, 2^2$)	0.0322 46.74 37 ($10^0, 2^2$)	0.0123 46.52 36 ($10^4, 10^1, 2^2$)	0.0133 49.05 37 ($10^1, 10^3, 2^3$)	0.0121 53.08 36 ($10^4, 10^3, 2^5$)
Tae (150 × 5, 3)	0.0047 32.68 37 -	0.0033 54.67 36 -	0.021 32.68 37 ($10^{-5}, 2^{-5}$)	0.0206 42.13 35 ($10^1, 2^2$)	0.0067 47.53 37 ($10^0, 10^4, 2^3$)	0.0074 32.68 37 ($10^{-5}, 10^{-5}, 2^{-5}$)	0.0064 54.88 37 ($10^4, 10^2, 2^5$)
Hayes-roth (132 × 4, 3)	0.0088 62.75 27 -	0.0021 33.19 33 -	0.0199 63.92 29 ($10^{-5}, 2^{-1}$)	0.0201 63.92 29 ($10^{-5}, 2^{-1}$)	0.0068 52.96 33 ($10^{-3}, 10^{-1}, 2^2$)	0.0071 54.5 30 ($10^5, 10^{-4}, 2^1$)	0.0061 70.35 24 ($10^0, 10^2, 2^1$)
Shuttle (1486 × 9, 5)	0.0098 63.25 167 -	0.0026 68.37 145 -	0.0173 70.33 150 ($10^{-4}, 2^5$)	0.017 70.59 149 ($10^{-5}, 2^4$)	0.0051 75.73 137 ($10^1, 10^4, 2^5$)	0.0056 64.66 160 ($10^{-3}, 10^{-5}, 2^2$)	0.0049 79.96 132 ($10^{-1}, 10^4, 2^5$)
Libras (360 × 90, 15)	0.3162 70.99 89 -	0.565 71.75 83 -	2.5814 85.68 88 ($10^{-5}, 2^{-1}$)	2.5527 81.56 86 ($10^{-3}, 2^0$)	1.1689 86.41 88 ($10^3, 10^2, 2^{-1}$)	2.168 88.6 87 ($10^{-4}, 10^{-5}, 2^0$)	1.1518 87.15 88 ($10^4, 10^0, 2^4$)
	0.4334 -	0.1532 -	0.251 -	0.2675 -	0.0799 -	0.08 -	0.074 -
Average rank (Accuracy)	6.1739	4.8696	4.3478	3.7391	3.4565	3.6739	1.7391
Average rank (# Miss)	5.8913	4.4348	4.4783	3.6957	3.6522	3.5435	2.3043
Win-Tie-Loss	23-0-0	21-1-1	19-2-2	19-2-2	18-1-4	17-2-4	

Table 1 also shows the number of mis-clustered [13] data points for every algorithm. The mis-clustered data points are calculated by counting the pair of data points with cluster mismatch. One can observe in Table 1 that even in terms of the mis-clustered data points, proposed LSPTSVC obtains the least rank i.e., 2.3696. Moreover, a similar trend is observed for the average ranks of other algorithms, in comparison to the average ranks based on accuracy. Only difference is in the comparison between FCM and kPC, where kPC is the worst performing algorithm in terms of mis-clustering. One can notice that for some datasets, the best performing algorithm in terms of accuracy is not having the least number of mis-clustered data points. This can be attributed to the imbalance in the number of data points of the clusters in a dataset.

The Win-Tie-Loss comparison is also shown in Table 1. The clustering accuracy of proposed LSPTSVC is compared with existing algorithms in a Win-Tie-Loss scenario for all the datasets. It is evident that LSPTSVC is having a 'Win' scenario for all the compared algorithms. The highest 'Win' case is in comparison to FCM, kPC, and TWSVC algorithm. This is because FCM algorithm is based on distance from neighbouring data points, while the datasets have varying data distributions. Moreover, LSPTSVC initializes its weights using the eigenvectors and then converges, while kPC obtains its hyperplanes as the eigenvectors. In comparison to TWSVC, proposed LSPTSVC involves the concept of within class scatter minimization leading to better clustering accuracy. However, proposed LSPTSVC is having some losses in case of TBSVC, LSTWSVC, and FLSTWSVC. For more analysis on significance of the proposed algorithm, statistical analysis is presented in Section 4.4.

The clusters identified by the proposed and existing algorithms using linear kernel for 3MC synthetic dataset are shown in Fig. 2. The actual clusters in the dataset are shown in Fig. 2a. One can easily notice that the clusters labelled by the proposed LSPTSVC in Fig. 2f are similar to the original clusters in Fig. 2a. This may be attributed to minimization of the within class variance of projected data, while keeping the projected data of other classes at unit distance from centre of the cluster.

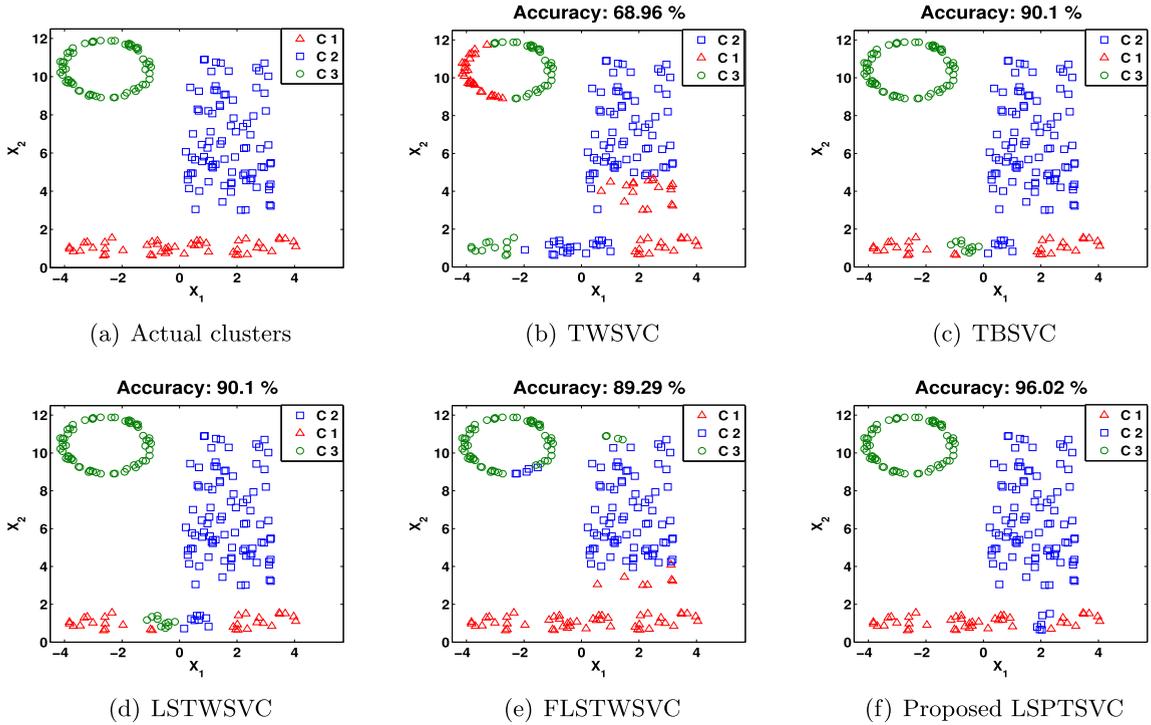


Fig. 2. Plot showing performance of proposed LSPTSVC in comparison to existing algorithms using linear kernel for 3MC dataset. In the legend ‘C’ means cluster.

Table 2 shows the performance of the proposed LSPTSVC for non-linear case. One can notice that clustering accuracy of the proposed LSPTSVC is better than existing algorithms for most of the datasets. This is also evident from the average rank of LSPTSVC which is the lowest among algorithms in Table 2 i.e., 1.7391. Moreover, the proposed non-linear LSPTSVC is having ‘Win’ situation with all the algorithms in most datasets. This is due to the effect of Gaussian kernel resulting in non-linear projection axes. Similar to the linear case, the training time of proposed LSPTSVC is also lesser than existing algorithms.

In terms of mis-clustered data points also, proposed LSPTSVC performs better than the existing algorithms with an average rank of 2.3043 (Table 2). A similar trend is observed for the average ranks of the different algorithms. However, the average rank of FLSTWSVC is lesser than LSTWSVC in terms of mis-clustered data points.

The clusters labelled by all the algorithms using non-linear kernel for Longsquare synthetic dataset are shown in Fig. 3. There are 6 clusters in this dataset labelled using non-linear kernel. It is clear from the illustration in Fig. 3f that LSPTSVC is able to label the clusters better than the other algorithms. Also, FLSTWSVC is showing similar performance in Fig. 3e. A similar trend is visible in Fig. 4 for Pathbased dataset, where proposed LSPTSVC outperforms the other algorithms.

4.4. Statistical analysis

In this section, we check the statistical significance of proposed LSPTSVC with existing techniques in terms of clustering accuracy. We perform the Friedman test [8] with the corresponding Nemenyi post hoc test. Initially, we assume that there is no difference between the methods as the null hypothesis.

4.4.1. Linear case

The χ^2_F value for Friedman test is calculated using Table 1 as

$$\chi^2_F = \frac{12N}{k(k+1)} \left[\sum_{i=1}^k R_i^2 - \frac{k(k+1)^2}{4} \right],$$

where R_i is the average rank on N datasets for i^{th} method.

$$\chi^2_F = \frac{12 \times 23}{7(7+1)} \left[\sum_{i=1}^7 R_i^2 - \frac{7(7+1)^2}{4} \right],$$

$$\chi^2_F = \frac{12 \times 23}{7(7+1)} \left[(5.8043^2 + 5.4783^2 + 4.1522^2 + 3.1522^2 + 3.6304^2 + 3.7609^2 + 2.0217^2) - \frac{7(7+1)^2}{4} \right],$$

$\approx 50.7162.$

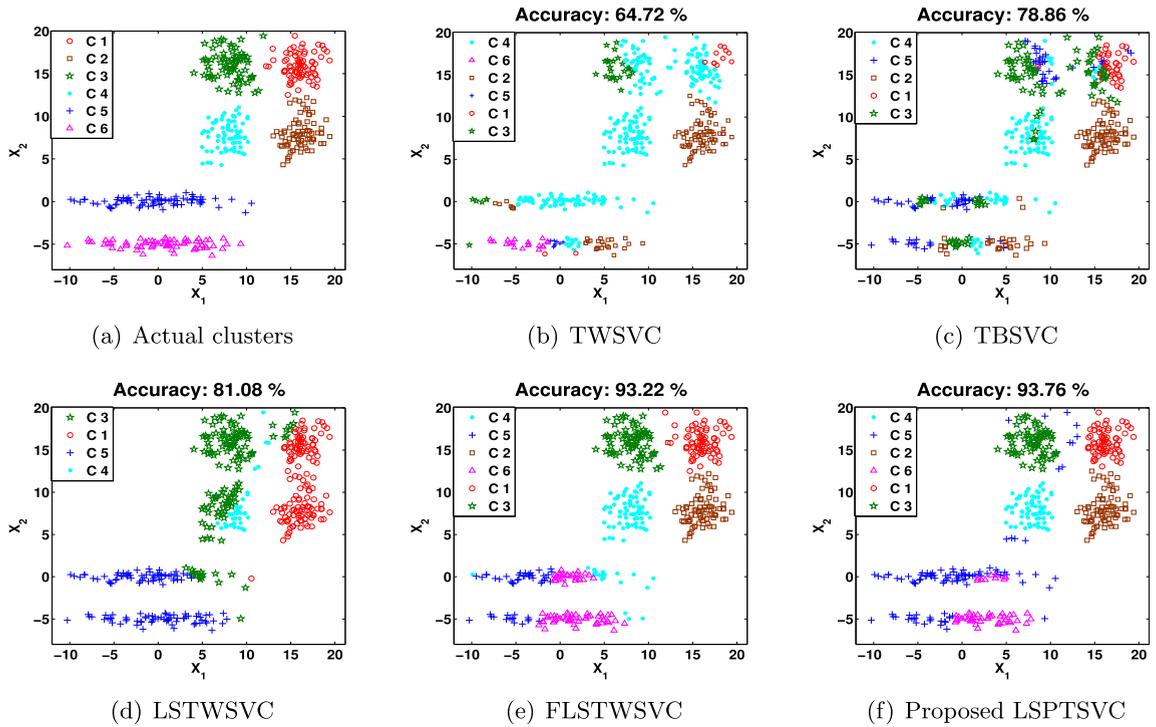


Fig. 3. Plot showing performance of proposed LSPTSVC in comparison to existing algorithms using Gaussian kernel for Longsquare dataset. In the legend 'C' means cluster.

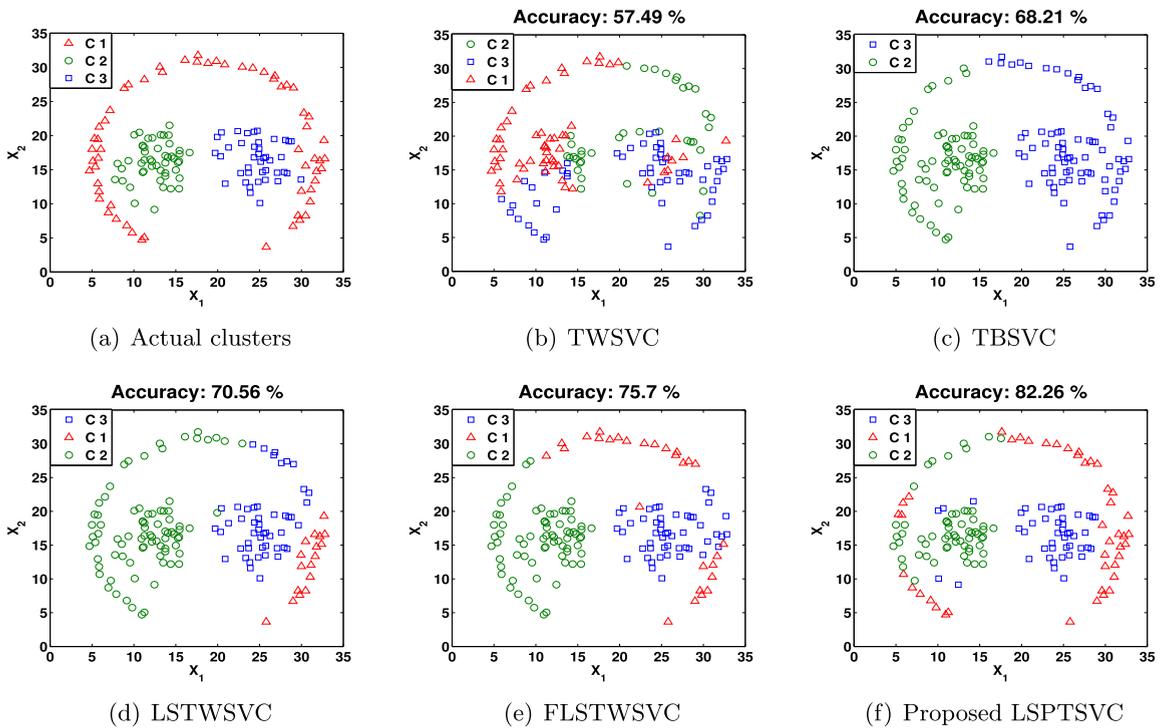


Fig. 4. Plot showing performance of proposed LSPTSVC in comparison to existing algorithms using Gaussian kernel for Pathbased dataset. In the legend 'C' means cluster.

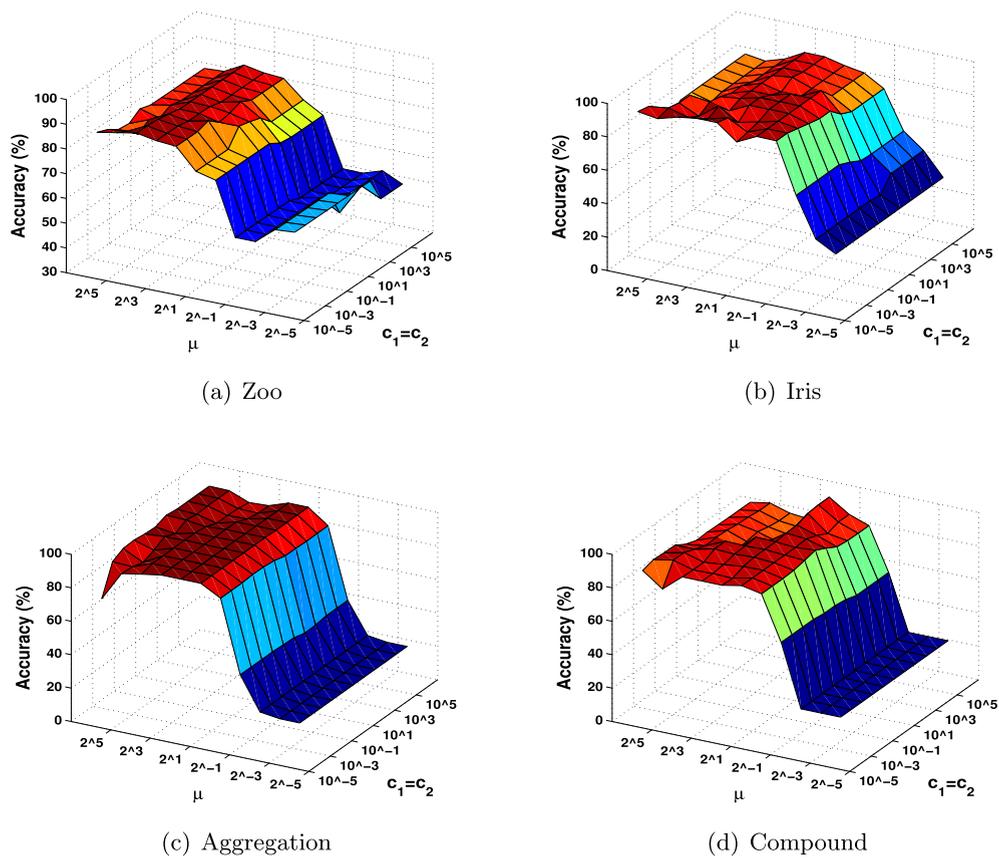


Fig. 5. Insensitivity of proposed LSPTSVC for clustering to the user specified parameters $c_1 = c_2$ and μ using Gaussian kernel for real world benchmark datasets.

Table 4

Performance comparison on accuracy (%) and training time of proposed LSPTSVC with existing algorithms on large sample datasets using linear kernel. Average rank is based on accuracy.

Dataset (Train size, Test size)	Clusters	LSTWSVC Accuracy Time (s)	FLSTWSVC Accuracy Time (s)	Proposed LSPTSVC Accuracy Time (s)
Pendigits (5996 × 17, 1498 × 17)	10	84.76 1.3836	79.77 1.5956	83.31 1.0081
Penbased (8794 × 16, 2198 × 16)	10	75.62 2.6765	71.7 2.9576	81.59 2.0746
Letter_10k (8000 × 16, 2000 × 16)	26	86.21 6.2636	86.64 6.9988	91.51 5.1504
Letter_20k (16000 × 16, 4000 × 16)	26	81.16 23.0756	87.54 24.9343	90.17 19.9956
Poker_30k (24000 × 10, 6000 × 10)	8	54.71 13.3279	53.28 16.7914	54.68 10.6532
Poker_40k (32000 × 10, 8000 × 10)	9	54.78 30.5703	53.98 36.0595	55.29 22.2977
Poker_50k (40000 × 10, 10000 × 10)	9	54.41 48.7161	54.51 62.3994	54.99 36.6232
Average rank		2.1429	2.5714	1.2857

of features. On TTC-3600 dataset, the training time of LSPTSVC is significantly lesser than the other algorithms. Also, the accuracy of LSPTSVC is better for all the datasets. However, the differences in training time of the algorithms are not high. This is due to inclusion of time for generation of kernel matrices in all the algorithms. The time required for generation of kernel matrices is very high in comparison to other steps in the algorithms.

Table 5

Performance comparison on accuracy (%) and training time of proposed LSPTSVC with existing algorithms on large feature datasets using Gaussian kernel. Average rank is based on accuracy.

Dataset (Train size, Test size)	Clusters	TSVC	TBSVC	LSTWSVC	FLSTWSVC	Proposed LSPTSVC Accuracy Time (s)
		Accuracy Time (s)	Accuracy Time (s)	Accuracy Time (s)	Accuracy Time (s)	
Dbworld_emails (52 × 4702, 12 × 4702)	2	83.33	69.7	69.7	69.7	100
		0.075	0.0716	0.0611	0.0669	0.0598
Hydraulic_condition_ps1 (1103 × 6000, 1102 × 6000)	2	50.03	62.06	50.93	59.21	62.97
		61.3294	61.5112	60.8006	60.8788	60.6945
Hydraulic_condition_ps2 (1103 × 6000, 1102 × 6000)	2	59.85	52.45	56.03	53.52	70.85
		60.6696	61.1182	60.4428	60.5068	60.2428
Hydraulic_condition_ps4 (1544 × 6000, 661 × 6000)	2	66.39	66.74	67.63	62.33	68.91
		119.679	119.708	118.259	118.567	117.561
TTC-3600 (2880 × 7507, 720 × 7507)	6	68.54	66.76	65.09	60.59	69.31
		687.917	660.138	632.627	632.861	613.667
Average rank		3	3.4	3.4	4.2	1

4.6. Applications

In this section, we present some real world applications of the proposed LSPTSVC, along with comparisons to existing algorithms. Experiments are performed on biometric data viz. facial, and facial expression images. Moreover, we also present the application of LSPTSVC on biomedical data. We use ADNI MRI data to evaluate clustering ability of LSPTSVC on Alzheimer's disease. For all the applications, Gaussian kernel is used in proposed and existing algorithms.

4.6.1. Face clustering

A total of 400 images are included from the AT&T face recognition database shown in Fig. 6. The dataset consists of 10 images of 40 individuals, each having dimension of 112×92 . 200 images are used in the training as well as testing phase. The dataset is constructed by using all pixel values of an image in one row of the dataset matrix. To avoid overfitting of model, we use principal component analysis (PCA) and class discriminatory ratio (CDR) [30] to reduce the dimension of the dataset to 400×100 . The results for face clustering are shown in Fig. 7. It is evident that LSPTSVC is able to cluster facial data with better accuracy i.e., 95.53% in comparison to other algorithms.



Fig. 6. AT&T face recognition data (AT&T Laboratories Cambridge) comprising of 40 individuals.

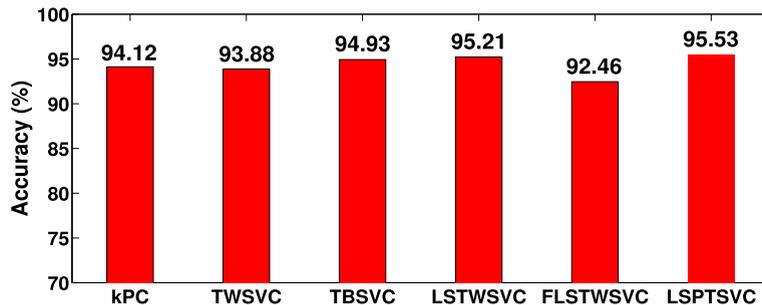


Fig. 7. Performance comparison of the proposed LSPTSVC with existing algorithms for clustering of AT&T face recognition data.

4.6.2. Facial expression clustering

A total of 210 images are downloaded from the well known JAFFE facial expression database [21] having 30 images of each expression. The dataset contains 7 classes as shown in Fig. 8. A total of 140 images are used for training consisting of 20 images of each class, and 70 for testing containing 10 images of each class. The dimension of all the images is 256×256 .

In comparison to face recognition, facial expression is a much more difficult problem due to large variations in facial expressions among individuals [19]. However, edge based information is useful for identifying facial expressions [12]. Wavelet transform is used for extracting high frequency components [30] responsible for the edges. Therefore, we used PCA along with wavelet transform for dimension reduction, and extraction of useful information for expression detection. Wavelet transform [30] is performed using ‘Daeubechies-4’ wavelet up to 3 levels of decomposition. Further, CDR is utilized to select the prominent features. After dimension reduction, the size of the dataset becomes 210×50 . The results for clustering are shown in Fig. 9. In comparison to face recognition, the accuracy is lower in all the algorithms for both features. However, it can be observed that the proposed LSPTSVC is showing highest clustering accuracy for both feature sets i.e. PCA and wavelet for the facial expression dataset.

Other algorithms obtaining high clustering accuracy are TBSVC and FLSTWSVC. This is because TBSVC involves the regularization term, and FLSTWSVC includes fuzzy membership information about the data points.

4.6.3. Alzheimer's disease clustering

Alzheimer's disease is an incurable disease affecting 50 million people worldwide [23]. Classification of Alzheimer's disease data is a challenging task [40]. For unlabelled Alzheimer data, clustering is a very useful option. As per our search, there is no work on application of SVM based clustering techniques for Alzheimer's disease data. Therefore, we used Alzheimer's data for clustering by the proposed LSPTSVC and compared with other algorithms.

150 T1-weighted structural MRI images are downloaded from the ADNI database. There are 50 images belonging to each of the three categories i.e., control normal (CN), mild cognitive impairment (MCI), and Alzheimer's disease (AD) as shown in Fig. 10. The average age of the subjects is 75.83 ± 6.07 years, and the Mini-Mental State Examination (MMSE) score of subjects is 26.51 ± 2.88 . The images are of the following specifications: field strength = 1.5 T; description: MP-RAGE, acquisition: 3D, acquisition plane: sagittal, pulse sequence: RM; slice thickness: 1.2 mm; flip angle: 8 degrees; manufacturer: GE medical systems.

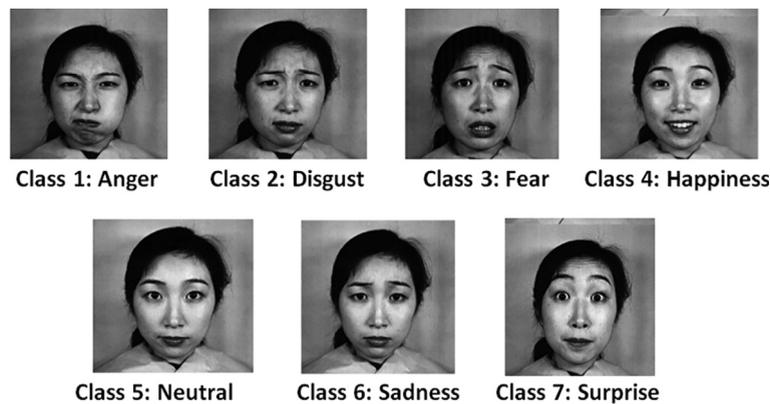


Fig. 8. Sample images of JAFFE database showing different facial emotions.

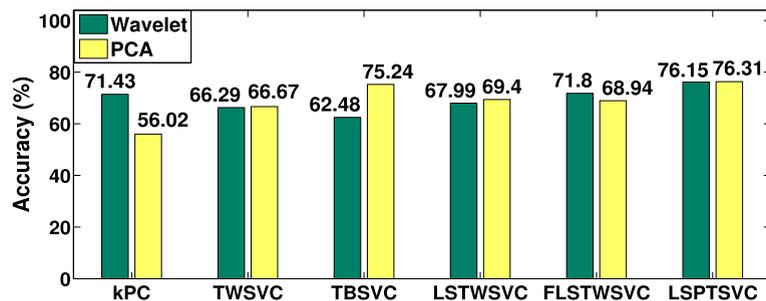


Fig. 9. Performance comparison of proposed LSPTSVC with existing algorithms for clustering of JAFFE facial expression data.

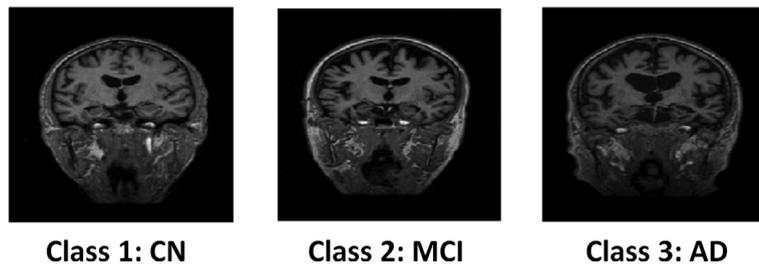


Fig. 10. MRI images of CN, MCI, and AD subject from ADNI database.

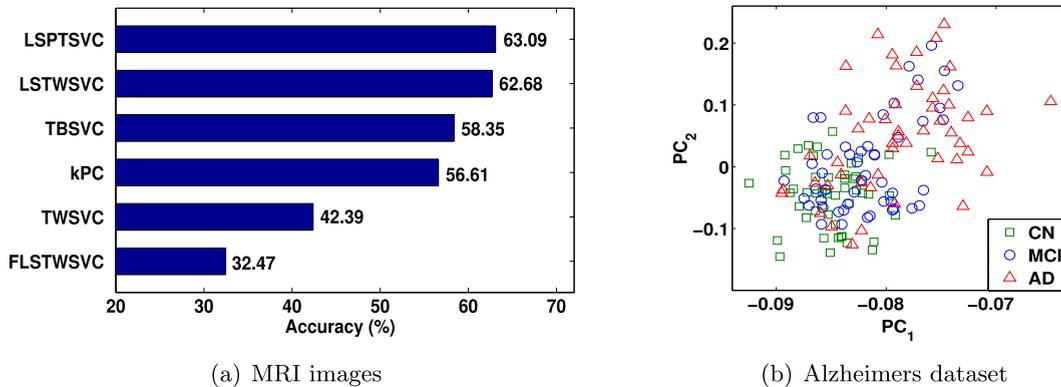


Fig. 11. (a) Performance comparison of proposed LSPTSVC with existing algorithms for clustering of ADNI Alzheimer's disease data. (b) Plot of Alzheimer's dataset using two PCA components.

The images are processed using Freesurfer to extract the volumetric and thickness measures of the brain. Since, one MCI image failed to process in the Freesurfer pipeline, the dimension of the dataset is 149×91 . The feature set includes 34 sub-cortical volumes, 23 WM tissue volumes, and 34 cortical thickness measures [32]. 74 images are used for training and 75 for testing. The results are shown in Fig. 11a. It is evident that the proposed LSPTSVC is effective in clustering Alzheimer's data in comparison to other algorithms. LSPTSVC obtains a clustering accuracy of 63.09% for CN, MCI and AD subjects. The clustering accuracy of LSPTSVC is similar to previous works [36,18] on multiclass classification of Alzheimer's data. However, the clustering accuracy of FLSTWSVC is lowest among all the methods. This may be attributed to the presence of outliers in the data. Indeed, the clustering accuracies of all the algorithms are comparatively lower than the applications discussed in the previous subsections. This is because the data points belonging to the classes MCI and AD are non-linear in their distribution, and are overlapping in nature [32] as shown in Fig. 11b. Moreover, MCI is an intermediate stage between CN and AD, leading to incorrect labelling of data points [46]. Therefore, the Alzheimer's dataset is difficult to classify [40] or cluster.

5. Conclusions and future work

In this work, we proposed a novel projection based clustering algorithm i.e., LSPTSVC. Proposed LSPTSVC finds projection axes instead of projection planes for clustering. This is an alternative to the plane based clustering algorithms. The solution of proposed LSPTSVC is obtained by solving a set of linear equations, leading to lesser computational cost. Consequently, no optimization toolbox is required for LSPTSVC. Experimental results show that proposed LSPTSVC obtains better clustering accuracy than existing algorithms with lesser training time. Statistical analysis also implies that the proposed algorithm is significantly better than existing algorithms. Moreover, LSPTSVC is an efficient algorithm for clustering on datasets with large sample and feature size. In future, the proposed LSPTSVC can be extended for multiple projection axes.

In case of real world applications, LSPTSVC performed better than the existing algorithms. This justifies its applicability for real world applications. In case of Alzheimer's disease, proposed LSPTSVC has shown significantly better performance, justifying its use for healthcare applications. In future, proposed LSPTSVC can be applied on other real world clustering problems.

CRedit authorship contribution statement

B. Richhariya: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing, Visualization. **M. Tanveer:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We are thankful to the anonymous reviewers for their constructive comments for improvement of the paper.

This work was supported by Science & Engineering Research Board (SERB) under Ramanujan fellowship Grant No. SB/S2/RJN-001/2016 and Early Career Research Award Grant No. ECR/2017/000053. It is also supported by Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA under Extra Mural Research (EMR) Scheme Grant No. 22(0751)/17/ EMR-II. We gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support. A sincere thanks to the Indian Institute of Technology Indore for providing Institute fellowship to Mr. Bharat Richhariya. The collection of data and sharing of this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904), and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). The funding for ADNI is provided by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. The dissemination of ADNI data is carried out by the Laboratory for Neuro Imaging at the University of Southern California.

References

- [1] L. Bai, Y.H. Shao, Z. Wang, C.N. Li, Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding, *Knowl.-Based Syst.* 163 (2019) 227–240.
- [2] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, *Comput. Geosci.* 10 (2–3) (1984) 191–203.
- [3] P.S. Bradley, O.L. Mangasarian, K-plane clustering, *J. Global Optim.* 16 (1) (2000) 23–32.
- [4] S.G. Chen, X.J. Wu, A new fuzzy twin support vector machine for pattern classification, *Int. J. Mach. Learn. Cybern.* 9 (9) (2018) 1553–1564.
- [5] X. Chen, J. Yang, Q. Ye, J. Liang, Recursive projection twin support vector machine via within-class variance minimization, *Pattern Recogn.* 44 (10–11) (2011) 2643–2655.
- [6] P.M. Cheung, J.T. Kwok, A regularization framework for multiple-instance learning, in: *Proceedings of the 23rd International Conference on Machine Learning*, ACM, 2006, pp. 193–200.
- [7] C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.* 20 (3) (1995) 273–297.
- [8] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (Jan) (2006) 1–30.
- [9] D. Dua, C. Graff, UCI machine learning repository, 2017. .
- [10] X. Gao, L. Fan, H. Xu, Multiple rank multi-linear kernel support vector machine for matrix data classification, *Int. J. Mach. Learn. Cybern.* 9 (2) (2018) 251–261.
- [11] G.H. Golub, C.F. Van Loan, *Matrix computations*, vol. 3, JHU Press, 2012.
- [12] R.J. Harris, A.W. Young, T.J. Andrews, Brain regions involved in processing facial identity and expression are differentially selective for surface and edge information, *NeuroImage* 97 (2014) 217–223.
- [13] L. Huang, D. Yan, N. Taft, M.I. Jordan, Spectral clustering with perturbed data, in: *Advances in Neural Information Processing Systems*, pp. 705–712. .
- [14] A.K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recogn. Lett.* 31 (8) (2010) 651–666.
- [15] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (5) (2007) 905–910.
- [16] R. Khemchandani, A. Pal, S. Chandra, Fuzzy least squares twin support vector clustering, *Neural Comput. Appl.* 29 (2) (2018) 553–563.
- [17] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, *Expert Syst. Appl.* 36 (4) (2009) 7535–7543.
- [18] R.K. Lama, J. Gwak, J.S. Park, S.W. Lee, Diagnosis of Alzheimer's disease based on structural MRI images using a regularized extreme learning machine and PCA features, *J. Healthcare Eng.* 2017 (2017), <https://doi.org/10.1155/2017/5485080>.
- [19] H. Li, G. Wen, Sample awareness-based personalized facial expression recognition, *Appl. Intell.* 49 (8) (2019) 2956–2969.
- [20] X. Liu, T. Zhu, L. Zhai, J. Liu, Mass classification of benign and malignant with a new twin support vector machine joint $l_{2,1}$ -norm, *Int. J. Mach. Learn. Cybern.* 10 (1) (2019) 155–171.
- [21] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 1998, pp. 200–205.
- [22] S. Ma, B. Cheng, Z. Shang, G. Liu, Scattering transform and LSPTSVM based fault diagnosis of rotating machinery, *Mech. Syst. Signal Process.* 104 (2018) 155–170.
- [23] C. Patterson, The state of the art of dementia research: New frontiers, *World Alzheimer's Report 2018* (2018).
- [24] X. Peng, D. Chen, PTSVRs: regression models via projection twin support vector machine, *Inf. Sci.* 435 (2018) 1–14.
- [25] Z. Qi, Y. Tian, Y. Shi, Robust twin support vector machine for pattern classification, *Pattern Recogn.* 46 (1) (2013) 305–316.
- [26] J. Qiu, K. Sun, I.J. Rudas, H. Gao, Command filter-based adaptive nn control for mimo nonlinear systems with full-state constraints and actuator hysteresis, *IEEE Trans. Cybern.* (2019), <https://doi.org/10.1109/TCYB.2019.2944761>.
- [27] J. Qiu, K. Sun, T. Wang, H. Gao, Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance, *IEEE Trans. Fuzzy Syst.* 27 (11) (2019) 2152–2162.
- [28] R. Rastogi, A. Pal, Fuzzy semi-supervised weighted linear loss twin support vector clustering, *Knowl.-Based Syst.* 165 (2019) 132–148.

- [29] M. Reuter, N.J. Schmansky, H.D. Rosas, B. Fischl, Within-subject template estimation for unbiased longitudinal image analysis, *NeuroImage* 61 (4) (2012) 1402–1418.
- [30] B. Richhariya, D. Gupta, Facial expression recognition using iterative universum twin support vector machine, *Appl. Soft Comput.* 76 (2019) 53–67.
- [31] B. Richhariya, M. Tanveer, EEG signal classification using universum support vector machine, *Expert Syst. Appl.* 106 (2018) 169–182.
- [32] B. Richhariya, M. Tanveer, A.H. Rashid, Alzheimer's Disease Neuroimaging Initiative, Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE), *Biomed. Signal Process. Control* 59 (2020) 101903.
- [33] Y.H. Shao, L. Bai, Z. Wang, X.Y. Hua, N.Y. Deng, Proximal plane clustering via eigenvalues, *Proc. Comput. Sci.* 17 (2013) 41–47.
- [34] Y.H. Shao, N.Y. Deng, Z.M. Yang, Least squares recursive projection twin support vector machine for classification, *Pattern Recogn.* 45 (6) (2012) 2299–2307.
- [35] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, *IEEE Trans. Neural Networks* 22 (6) (2011) 962–968.
- [36] L. Sørensen, C. Igel, A. Pai, I. Balas, C. Anker, M. Lillholm, M. Nielsen, Alzheimer's Disease Neuroimaging Initiative, Differential diagnosis of mild cognitive impairment and Alzheimer's disease using structural MRI cortical thickness, hippocampal shape, hippocampal texture, and volumetry, *NeuroImage: Clinical* 13 (2017) 470–482.
- [37] K. Sun, S. Mou, J. Qiu, T. Wang, H. Gao, Adaptive fuzzy control for nontriangular structural stochastic switched nonlinear systems with full state constraints, *IEEE Trans. Fuzzy Syst.* 27 (8) (2018) 1587–1601.
- [38] M. Tanveer, C. Gautam, P. Suganthan, Comprehensive evaluation of twin svm based classifiers on uci datasets, *Appl. Soft Comput.* 83 (2019) 105617.
- [39] M. Tanveer, M.A. Khan, S.S. Ho, Robust energy-based least squares twin support vector machines, *Appl. Intell.* 45 (1) (2016) 174–186.
- [40] M. Tanveer, B. Richhariya, R.U. Khan, A.H. Rashid, P. Khanna, M. Prasad, C.T. Lin, Machine learning techniques for the diagnosis of Alzheimer's disease: a review, *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)* 16 (1s) (2020) 1–35.
- [41] M. Tanveer, A. Sharma, P.N. Suganthan, General twin support vector machine with pinball loss function, *Inf. Sci.* 494 (2019) 311–327.
- [42] M. Tanveer, K. Shubham, M. Aldhaifallah, K. Nisar, An efficient implicit regularized lagrangian twin support vector regression, *Appl. Intell.* 44 (4) (2016) 831–848.
- [43] Y. Tian, Z. Qi, Review on: twin support vector machines, *Ann. Data Sci.* 1 (2) (2014) 253–277.
- [44] S. Wan, M.W. Mak, Predicting subcellular localization of multi-location proteins by improving support vector machines with an adaptive-decision scheme, *Int. J. Mach. Learn. Cybern.* 9 (3) (2018) 399–411.
- [45] Z. Wang, Y.H. Shao, L. Bai, N.Y. Deng, Twin support vector machine for clustering, *IEEE Trans. Neural Networks Learn. Syst.* 26 (10) (2015) 2583–2588.
- [46] Z. Wang, Y. Zheng, D.C. Zhu, A.C. Bozoki, T. Li, Classification of Alzheimer's disease, mild cognitive impairment and normal control subjects using resting-state fMRI based network connectivity analysis, *IEEE J. Transl. Eng. Health Med.* 6 (2018) 1–9.
- [47] E. Westman, J.S. Muehlboeck, A. Simmons, Combining MRI and CSF measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion, *NeuroImage* 62 (1) (2012) 229–238.
- [48] Z.M. Yang, H.J. Wu, C.N. Li, Y.H. Shao, Least squares recursive projection twin support vector machine for multi-class classification, *Int. J. Mach. Learn. Cybern.* 7 (3) (2016) 411–426.
- [49] A.L. Yuille, A. Rangarajan, The concave-convex procedure (CCCP), in: *Advances in Neural Information Processing Systems*, 2002, pp. 1033–1040.
- [50] N.N. Zhao, X.Y. Ouyang, C. Gao, L. Wang, A v-twin projection SVR with automatic accuracy adjustment, *Artif. Intell. Rev.* 53 (2020) 1511–1527.