



A fuzzy universum least squares twin support vector machine (FULSTSVM)

B. Richhariya¹ · M. Tanveer¹ · for the Alzheimer's Disease Neuroimaging Initiative

Received: 20 August 2020 / Accepted: 7 January 2021

© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

Abstract

Universum based twin support vector machines give prior information about the distribution of data to the classifier. This leads to better generalization performance of the model, due to the universum. However, in many applications the data points are not equally useful for the classification task. This leads to the use of fuzzy membership functions for the datasets. Similarly, in universum based algorithms, all the universum data points are not equally important for the classifier. To solve these problems, a novel fuzzy universum least squares twin support vector machine (FULSTSVM) is proposed in this work. In FULSTSVM, the membership values are used to provide weights for the data samples of the classes, as well as to the universum data. Further, the optimization problem of proposed FULSTSVM is obtained by solving a system of linear equations. This leads to an efficient fuzzy based algorithm. Numerical experiments are performed on various benchmark datasets, with discussions on generalization performance, and computational cost of the algorithms. The proposed FULSTSVM outperformed the existing algorithms on most datasets. A comparison is presented for the performance of the proposed and other baseline algorithms using statistical significance tests. To show the applicability of FULSTSVM, applications are also presented, such as detection of Alzheimer's disease, and breast cancer.

Keywords Universum · Fuzzy membership · Least squares twin support vector machine · Outliers · Alzheimer's disease

1 Introduction

Support vector machines (SVMs) are one of the widely used supervised learning algorithms for classification problems [5]. SVM has been applied on applications ranging from face recognition [36], text categorization [30], to diagnosis of diseases, such as epilepsy [21, 35] or

Alzheimer's disease [28]. The objective function of SVM is convex, leading to a quadratic programming problem (QPP). Moreover, due to the convexity, the optimization problem of SVM gives a globally optimal solution. This is the benefit of SVM in comparison to techniques such as artificial neural networks (ANN), which give rise to locally optimal solutions. However, for solving a QPP, the computational complexity is very high i.e. $O(m^3)$, where m is the number of samples. In order to reduce the overhead of solving one large QPP, Jayadeva et al. [13] proposed a twin support vector machine (TWSVM). TWSVM is based on the idea of twin hyperplanes, rather than one as in SVM. Further, Kumar and Gopal [15] proposed a least squares twin SVM (LSTSVM) by minimizing the squared errors of the data points. The computation time of LSTSVM is very less in comparison to TWSVM. However, the LSTSVM classifier is more sensitive to outliers, due to the quadratic loss function. In an evaluation of 187 classifiers, robust energy-based least squares twin support vector machine (RELSTSVM) [27] turned out to be one of the better performing classifiers. This can be attributed to the energy values used for removing the effect of noise.

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

✉ M. Tanveer
mtanveer@iiti.ac.in

B. Richhariya
phd1701241001@iiti.ac.in

¹ Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India

Weston et al. [32] proposed the idea of universum data to improve the generalization performance of the SVM classifier. Universum is introduced in SVM based on the principle of maximal contradiction. In the formulation of SVM, there is no information about data distribution. Therefore, by introducing the universum data, prior information is incorporated in the optimization problem of SVM. This prior information is in a form of Bayesian prior for SVM [32]. Unlike the data points belonging to the classes, the universum data are unlabelled, and lie in between the binary classes. The universum based SVM algorithm (USVM) includes the universum points in an ϵ -insensitive tube between the binary classes. This algorithm is further improved in [18] by proposing a universum based twin support vector machine (UTSVM). The UTSVM algorithm computes the hyperplanes faster than USVM. To reduce the computation cost of UTSVM, Xu et al. [33] proposed a universum least squares twin support vector machine (ULSTSVM) by using quadratic loss for universum points. However, in many applications, the generation of universum is not proper for the classification problem [21], and is an active area of research. Cherkassky et al. [6] discussed about the conditions for the effectiveness of universum data. Like other forms of data, the universum data also suffer from the problem of noise, and therefore proper weights must be given to the data points of the classes, and the universum.

In many applications, the data are generated with noise in the features or labels. One way to solve this is to use fuzzy memberships for the different data points. Lin and Wang [16] proposed a fuzzy SVM using distance based fuzzy functions. Fuzzy SVM has also been used for multi-class problems [30], using a one-against-all (OAA) approach. This approach of fuzzy memberships is extended to class imbalance problems by Batuwita and Palade [4] using different fuzzy membership functions. Recently, an efficient fuzzy based approach is proposed as robust least squares twin support vector machine for class imbalance learning (RFLSTSVM-CIL) [22]. In the RFLSTSVM-CIL algorithm, a fuzzy membership function is also proposed to remove the effect of noisy data in class imbalance scenarios. A bilateral-weighted fuzzy SVM classifier is proposed [3] by assigning fuzzy membership values to the data points for both the classes. For detection of breast cancer, hepatitis, and diabetes, a weighted least squares twin SVM (WLSTSVM) [29] is proposed using hybrid feature selection. Moreover, a weighted least squares SVM is used with manifold regularization for nonlinear systems [19].

For universum based algorithms, an entropy based fuzzy membership is used in [23] to propose a fuzzy universum SVM (FUSVM) and fuzzy universum twin SVM (FUTSVM). In FUSVM and FUTSVM, the universum data is given weightage on the basis of fuzzy membership

values. Fuzzy based twin SVMs have also been used in financial risk predictions. A kernel based fuzzy twin SVM algorithm [12] is proposed for estimating financial risks. A novel twin SVM algorithm [11] is proposed using fuzzy hyperplane for stock price prediction using financial news articles. Moreover, in the case of incremental and decremental learning, a fuzzy twin bounded support vector machine is proposed [17]. Even for regression problems, fuzzy based SVM models have been proposed, such as for financial time series forecasting [14].

In this work, to show the applicability of the proposed algorithm, we have included two applications: one is a neurological disease i.e. Alzheimer's disease, and other is breast cancer. Alzheimer's disease is a neuro-degenerative disease, which is detected through magnetic resonance imaging (MRI). Usually, the patients affected are of age 60 years and above [9]. Breast cancer consists of the majority of cancers in woman [34]. Machine learning techniques can improve the classification of identifying the cancer. Therefore, we used histopathological images to classify breast cancer using the proposed approach.

Inspired by the approach of RFLSTSVM-CIL and universum learning, in this work we present a novel fuzzy based universum least squares twin support vector machine (FULSTSVM). The main contributions of this work are as follows:

- A novel fuzzy based least squares twin SVM algorithm is presented using universum data.
- To remove the impact of outliers, fuzzy memberships are calculated for the data points of the classes.
- Universum data are utilized to give prior information about data distribution.
- To give proper information about data distribution, fuzzy membership is assigned to the universum data points.
- The proposed fuzzy based approach is incorporated with a least squares model, leading to a system of linear equations for generating the classifier.
- Experiments are shown for benchmark datasets, with comparative analysis of the proposed and baseline algorithms.
- Applications are shown on Alzheimer's disease and breast cancer datasets.

We use the following mathematical notations in this work: A vector represented as w is a column vector. The matrices D_1 and D_2 contain the data points of the binary classes having size $m_1 \times n$ and $m_2 \times n$, respectively, where n is the number of features in each sample. The total number of data points are $m = m_1 + m_2$. The universum data matrix is denoted by U having dimension $m_3 \times n$. The transpose of a vector is denoted as w^t , 2-norm is represented by $\|w\|$, and e is a vector of ones of appropriate dimension.

2 Related work

This section presents the formulations of two related algorithms for this work in brief. The algorithms are twin support vector machine (TWSVM), and universum least squares twin support vector machine (ULSTSVM).

2.1 TWSVM

The twin hyperplanes of nonlinear TWSVM algorithm are generated by solving the following optimization problems,

$$\min_{w_1, b_1, \xi_1} \frac{1}{2} \|K(D_1, G^t)w_1 + e_1 b_1\|^2 + c_1 e_1^t \xi_1 \quad (1)$$

$$s.t. - (K(D_2, G^t)w_1 + e_2 b_1) + \xi_1 \geq e_2, \xi_1 \geq 0,$$

$$\min_{w_2, b_2, \xi_2} \frac{1}{2} \|K(D_2, G^t)w_2 + e_2 b_2\|^2 + c_2 e_2^t \xi_2 \quad (2)$$

$$s.t. K(D_1, G^t)w_2 + e_1 b_2 + \xi_2 \geq e_1, \xi_2 \geq 0,$$

where $K(D_1, G^t)$ is the kernel matrix, $G = [D_1; D_2]$, ξ_i is slack variable, c_i is penalty parameter, and e_i is vector of ones of suitable dimension, $i = 1, 2$.

The classifiers are generated by calculating the parameters w and b by solving Wolfe duals [13] of Eqs. (1) and (2), written as

$$\max_{\alpha_1} e_1^t \alpha_1 - \frac{1}{2} \alpha_1^t N (M^t M)^{-1} N^t \alpha_1 \quad (3)$$

$$s.t. 0 \leq \alpha_1 \leq c_1$$

and

$$\max_{\alpha_2} e_2^t \alpha_2 - \frac{1}{2} \alpha_2^t M (N^t N)^{-1} M^t \alpha_2 \quad (4)$$

$$s.t. 0 \leq \alpha_2 \leq c_2,$$

where

$$M = [K(D_1, G^t) e_1] \text{ and } N = [K(D_2, G^t) e_2]$$

; α_1 and α_2 are the vectors of Lagrange multipliers. The classifying hyperplanes $K(x^t, G^t)w_1 + b_1 = 0$ and $K(x^t, G^t)w_2 + b_2 = 0$, where x is a data point, are generated using the parameters $w_i (i = 1, 2)$ and $b_i (i = 1, 2)$ from the following equations,

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(M^t M + \delta I)^{-1} N^t \alpha_1, \quad (5)$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (N^t N + \delta I)^{-1} M^t \alpha_2, \quad (6)$$

where $\delta > 0$ is a small positive value for avoiding ill-conditioning of the matrices $M^t M$ and $N^t N$, and I is an identity matrix of suitable size.

Using the following decision function, the class is assigned to a new data point x .

$$class(x) = \arg \min_{i=1,2} |K(x, G^t)w_i + e_i b_i|. \quad (7)$$

2.2 ULSTSVM

The optimization problems of nonlinear ULSTSVM [18] are described as

$$\min_{w_1, b_1, \xi_1, \psi_1} \frac{1}{2} \|K(D_1, G^t)w_1 + e_1 b_1\|^2 + \frac{c_1}{2} \|\xi_1\|^2$$

$$+ \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_u}{2} \|\psi_1\|^2 \quad (8)$$

$$s.t. - (K(D_2, G^t)w_1 + e_2 b_1) + \xi_1 = e_2,$$

$$K(U, G^t)w_1 + e_u b_1 + \psi_1 = (-1 + \epsilon)e_u,$$

$$\min_{w_2, b_2, \xi_2, \psi_2} \frac{1}{2} \|K(D_2, G^t)w_2 + e_2 b_2\|^2$$

$$+ \frac{c_2}{2} \|\xi_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_u}{2} \|\psi_2\|^2 \quad (9)$$

$$s.t. K(D_1, G^t)w_2 + e_1 b_2 + \xi_2 = e_1,$$

$$- (K(U, G^t)w_2 + e_u b_2) + \psi_2 = (-1 + \epsilon)e_u,$$

where ξ_i, ψ_i , are the slack variables, c_i, c_u , $i = 1, 2$ are positive penalty parameters, and c_i , $i = 3, 4$ are positive parameters for the regularization.

Rewriting the optimization problem as an unconstrained problem using values of error variables,

$$\min_{w_1, b_1} \frac{1}{2} \|K(D_1, G^t)w_1 + e_1 b_1\|^2$$

$$+ \frac{c_1}{2} \|K(D_2, G^t)w_1 + e_2 b_1 + e_2\|^2$$

$$+ \frac{c_3}{2} (\|w_1\|^2 + b_1^2)$$

$$+ \frac{c_u}{2} \| - (K(U, G^t)w_1 + e_u b_1) + (-1 + \epsilon)e_u \|^2, \quad (10)$$

$$\min_{w_2, b_2} \frac{1}{2} \|K(D_2, G^t)w_2 + e_2 b_2\|^2$$

$$+ \frac{c_2}{2} \| - (K(D_1, G^t)w_2 + e_1 b_2) + e_1 \|^2$$

$$+ \frac{c_4}{2} (\|w_2\|^2 + b_2^2)$$

$$+ \frac{c_u}{2} \| (K(U, G^t)w_2 + e_u b_2) + (-1 + \epsilon)e_u \|^2. \quad (11)$$

By equating the gradient of Eq. (10) w.r.t. w_1 and b_1 equal to 0, we get

$$K(D_1, G^t)^t (K(D_1, G^t)w_1 + e_1 b_1)$$

$$+ c_1 K(D_2, G^t)^t (K(D_2, G^t)w_1 + e_2 b_1 + e_2)$$

$$+ c_3 w_1 + c_u K(U, G^t)^t (K(U, G^t)w_1$$

$$+ e_u b_1 - (-1 + \epsilon)e_u) = 0, \quad (12)$$

$$\begin{aligned}
 & e_1^t (K(D_1, G^t)w_1 + e_1b_1) \\
 & + c_1 e_2^t (K(D_2, G^t)w_1 + e_2b_1 + e_2) \\
 & + c_3 b_1 + c_u e_u^t (K(U, G^t)w_2 \\
 & + e_ub_1 - (-1 + \epsilon)e_u) = 0.
 \end{aligned} \quad (13)$$

Rewriting Eqs. (12) and (13) and solving, we get

$$\begin{aligned}
 [w_1 \ b_1]^t = & - (M^t M + c_1 N^t N + c_3 I + c_u O^t O)^{-1} \\
 & (c_1 N^t e_2 + c_u (1 - \epsilon) O^t e_u),
 \end{aligned} \quad (14)$$

where $M = [K(D_1, G^t) \ e_1]$, $N = [K(D_2, G^t) \ e_2]$, and $O = [K(U, G^t) \ e_u]$. In a similar manner, by performing the same procedure on Eq. (11), we get

$$\begin{aligned}
 [w_2 \ b_2]^t = & (N^t N + c_2 M^t M + c_4 I + c_u O^t O)^{-1} \\
 & (c_2 M^t e_1 + c_u (1 - \epsilon) O^t e_u).
 \end{aligned} \quad (15)$$

The class of a new data point is assigned based on the proximal hyperplane [33].

3 Proposed fuzzy universum least squares twin support vector machine (FULSTSVM)

This section presents the formulation of the proposed FULSTSVM algorithm in the linear and nonlinear form, with the fuzzy membership function. The proposed algorithm is motivated by the approach used in RFLSTSVM-CIL for removing the effect of outliers. In proposed FULSTSVM, the fuzzy memberships are calculated for the data samples belonging to the classes, as well as to the universum using fuzzy membership matrices as described below.

3.1 Linear FULSTSVM

The formulation of proposed FULSTSVM for the linear case is described using optimization problems (16) and (17). In the objective function of the primal problem (16), we use three diagonal matrices represented by S_i containing the fuzzy memberships of the data points of i^{th} class. The memberships of the data points are calculated on the basis of distance from their respective class centres.

We also add regularization in the objective function to include the structural risk minimization principle (SRM) principle. The constraints are similar to the ULSTSVM formulation described in the previous subsection. Figure 1 shows a pictorial representation of the proposed approach.

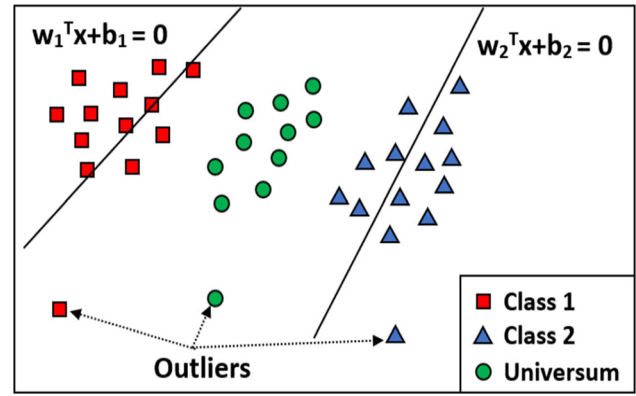


Fig. 1 Universum data with noise

$$\begin{aligned}
 \min_{w_1, b_1, \xi_1, \psi_1} & \frac{1}{2} \|S_1(D_1 w_1 + e_1 b_1)\|^2 \\
 & + \frac{c_1}{2} \|S_2 \xi_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_u}{2} \|S_u \psi_1\|^2
 \end{aligned} \quad (16)$$

$$\text{s.t. } -(D_2 w_1 + e_2 b_1) + \xi_1 = e_2,$$

$$U w_1 + e_u b_1 + \psi_1 = (-1 + \epsilon) e_u,$$

$$\begin{aligned}
 \min_{w_2, b_2, \xi_2, \psi_2} & \frac{1}{2} \|S_2(D_2 w_2 + e_2 b_2)\|^2 + \frac{c_2}{2} \|S_1 \xi_2\|^2 \\
 & + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_u}{2} \|S_u \psi_2\|^2
 \end{aligned} \quad (17)$$

$$\text{s.t. } D_1 w_2 + e_1 b_2 + \xi_2 = e_1,$$

$$-(U w_2 + e_u b_2) + \psi_2 = (-1 + \epsilon) e_u,$$

where S_i, S_u are diagonal matrices containing fuzzy membership values of data samples belonging to the classes and universum, respectively. ξ_i, ψ_i are the slack variables, and c_i, c_u are positive penalty parameters, $i = 1, 2$. The parameter for the insensitive zone is ϵ , while $c_i, i = 3, 4$ are the parameters for regularization.

Rewriting the objective functions using the values of the error variables,

$$\begin{aligned}
 \min_{w_1, b_1} & \frac{1}{2} \|S_1(D_1 w_1 + e_1 b_1)\|^2 \\
 & + \frac{c_1}{2} \|S_2(D_2 w_1 + e_2 b_1 + e_2)\|^2 \\
 & + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_u}{2} \|S_u(-(U w_1 + e_u b_1) + (-1 + \epsilon) e_u)\|^2,
 \end{aligned} \quad (18)$$

$$\begin{aligned}
 \min_{w_2, b_2} & \frac{1}{2} \|S_2(D_2 w_2 + e_2 b_2)\|^2 \\
 & + \frac{c_2}{2} \|S_1(-(D_1 w_2 + e_1 b_2) + e_1)\|^2 \\
 & + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_u}{2} \|S_u(U w_2 + e_u b_2 + (-1 + \epsilon) e_u)\|^2.
 \end{aligned} \quad (19)$$

By setting the gradient of QPP (18) w.r.t. w_1 and b_1 equal to 0, and solving we get

$$\begin{aligned}
 & c_3 w_1 + (S_1 D_1)^t (S_1 (D_1 w_1 + e_1 b_1)) \\
 & + c_1 (S_2 D_2)^t (S_2 (D_2 w_1 + e_2 b_1 + e_2)) \\
 & - c_u (S_u U)^t (S_u (- (U w_1 + e_u b_1) + (-1 + \epsilon) e_u)) = 0,
 \end{aligned} \quad (20)$$

$$\begin{aligned}
 & c_3 b_1 + (S_1 e_1)^t (S_1 (D_1 w_1 + e_1 b_1)) \\
 & + c_1 (S_2 e_2)^t (S_2 (D_2 w_1 + e_2 b_1 + e_2)) \\
 & - c_u (S_u e_u)^t (S_u (- (U w_1 + e_u b_1) + (-1 + \epsilon) e_u)) = 0,
 \end{aligned} \quad (21)$$

Rewriting Eqs. (20) and (21) with $u_1 = [w_1 \ b_1]^t$ and combining, we get

$$\begin{aligned}
 & c_3 u_1 + V^t V u_1 + c_1 W^t W u_1 + c_1 W^t S_2 e_2 \\
 & + c_u Z^t Z u_1 + c_u Z^t S_u (1 - \epsilon) e_u = 0
 \end{aligned} \quad (22)$$

Rearranging the terms and solving, we get

$$\begin{aligned}
 [w_1 \ b_1]^t &= - (V^t V + c_1 W^t W + c_3 I + c_u Z^t Z)^{-1} \\
 & (c_1 W^t S_2 e_2 + c_u Z^t S_u (1 - \epsilon) e_u),
 \end{aligned} \quad (23)$$

where $V = [S_1 D_1 \ S_1 e_1]$, $W = [S_2 D_2 \ S_2 e_2]$, and $Z = [S_u U \ S_u e_u]$.

Similarly, using the procedure for Eq. (19) and solving, we get

$$\begin{aligned}
 [w_2 \ b_2]^t &= (W^t W + c_2 V^t V + c_4 I + c_u Z^t Z)^{-1} \\
 & (c_2 V^t S_1 e_1 + c_u Z^t S_u (1 - \epsilon) e_u).
 \end{aligned} \quad (24)$$

A new data point x is classified using the following function,

$$\text{class}(x) = \arg \min_{i=1,2} \frac{|x^t w_i + e_i b_i|}{\|w_i\|}. \quad (25)$$

3.2 Nonlinear FULSTSVM

The formulation of nonlinear FULSTSVM is written as

$$\begin{aligned}
 & \min_{w_1, b_1, \xi_1, \psi_1} \frac{1}{2} \|S_1 (K(D_1, G^t) w_1 + e_1 b_1)\|^2 \\
 & + \frac{c_1}{2} \|S_2 \xi_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) \\
 & + \frac{c_u}{2} \|S_u \psi_1\|^2
 \end{aligned} \quad (26)$$

$$\begin{aligned}
 & s.t. \ - (K(D_2, G^t) w_1 + e_2 b_1) + \xi_1 = e_2, \\
 & K(U, G^t) w_1 + e_u b_1 + \psi_1 = (-1 + \epsilon) e_u,
 \end{aligned}$$

$$\begin{aligned}
 & \min_{w_2, b_2, \xi_2, \psi_2} \frac{1}{2} \|S_2 (K(D_2, G^t) w_2 + e_2 b_2)\|^2 \\
 & + \frac{c_2}{2} \|S_1 \xi_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) \\
 & + \frac{c_u}{2} \|S_u \psi_2\|^2
 \end{aligned} \quad (27)$$

$$\begin{aligned}
 & s.t. \ K(D_1, G^t) w_2 + e_1 b_2 + \xi_2 = e_1, \\
 & - (K(U, G^t) w_2 + e_u b_2) + \psi_2 = (-1 + \epsilon) e_u,
 \end{aligned} \quad (28)$$

where $K(D_1, G^t)$ is the kernel matrix, $G = [D_1; D_2]$, S_i, S_u are diagonal matrices containing fuzzy membership values of data points in the classes and universum, respectively, $i = 1, 2$.

Rewriting the objective functions using the constraints, we get

$$\begin{aligned}
 & \min_{w_1, b_1} \frac{1}{2} \|S_1 (K(D_1, G^t) w_1 + e_1 b_1)\|^2 \\
 & + \frac{c_1}{2} \|S_2 (K(D_2, G^t) w_1 + e_2 b_1 + e_2)\|^2 \\
 & + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) \\
 & + \frac{c_u}{2} \|S_u (- (K(U, G^t) w_1 + e_u b_1) + (-1 + \epsilon) e_u)\|^2,
 \end{aligned} \quad (29)$$

$$\begin{aligned}
 & \min_{w_2, b_2} \frac{1}{2} \|S_2 (K(D_2, G^t) w_2 + e_2 b_2)\|^2 \\
 & + \frac{c_2}{2} \|S_1 (- (K(D_1, G^t) w_2 + e_1 b_2) + e_1)\|^2 \\
 & + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_u}{2} \|S_u (K(U, G^t) w_2 \\
 & + e_u b_2 + (-1 + \epsilon) e_u)\|^2.
 \end{aligned} \quad (30)$$

The parameters w_1 and b_1 are obtained by setting the gradient of QPP (29) w.r.t. w_1 and b_1 equal to 0, and solving we get,

$$\begin{aligned}
 [w_1 \ b_1]^t &= - (M^t M + c_1 N^t N + c_3 I + c_u O^t O)^{-1} \\
 & (c_1 N^t S_2 e_2 + c_u O^t S_u (1 - \epsilon) e_u),
 \end{aligned} \quad (31)$$

where $M = [S_1 K(D_1, G^t) \ S_1 e_1]$, $N = [S_2 K(D_2, G^t) \ S_2 e_2]$, and $O = [S_u K(U, G^t) \ S_u e_u]$. Similarly, using Eq. (30), we get

$$\begin{aligned}
 [w_2 \ b_2]^t &= (N^t N + c_2 M^t M + c_4 I + c_u O^t O)^{-1} \\
 & (c_2 M^t S_1 e_1 + c_u O^t S_u (1 - \epsilon) e_u).
 \end{aligned} \quad (32)$$

For a new data point, similar to linear case, the class is assigned based on the class of the nearest hyperplane. In the following section, we present the fuzzy membership function used in the proposed FULSTSVM.

3.3 Fuzzy membership function

The proposed FULSTSVM utilizes a fuzzy function inspired by [16]. The following fuzzy function keeps the range of fuzzy memberships in the range (0.5, 1]. The membership function is described as

$$f(x_i) = 1 - 0.5 \left(\frac{|x_i - c_j|}{r_j + \delta} \right), \quad (33)$$

where x_i is a data point belonging to class j with centre c_j , $i = 1, \dots, m_j, j = 1, 2$. The variable r_j is the largest distance from the class centre of data points of class j , and δ is a very small positive value to avoid division by zero.

The range of fuzzy memberships in the above-mentioned fuzzy function is chosen as (0.5, 1]. This is to keep significant contribution of majority of the data points in the formation of the classifier. Moreover, the proposed FULSTSVM also gives fuzzy memberships to the universum data points. The contribution of most universum data points is required for providing prior information about the data, which is achieved by this function. Moreover, the contribution of outliers is reduced accordingly. This approach is in contrast to the approach proposed in FSVM [16], where the fuzzy memberships are chosen in the range (0, 1].

3.4 Time complexity

The time complexity of TWSVM is $2 * O(m/2)^3$ i.e. $O(m)^3/4$, where m is total number of data points [13]. This is the time complexity of solving the QPPs, which is a computationally intensive task. Moreover, TWSVM involves two matrix inverses having a complexity of $O(n)^3$, where n is the dimension of the matrix [2]. Similarly, UTWSVM has time complexity of $O(m + 2u)^3/4$, where u denotes the number of universum data points.

On the other hand, the formulation of LSTSVM involves solution of linear equations using two matrix inverses. Therefore, the computation time of LSTSVM is lesser than TWSVM and UTSVM in Table 1. Similarly, ULSTSVM involves two inverses with additional universum data. The time complexity of proposed FULSTSVM is similar to ULSTSVM, with additional complexity for fuzzy membership function. The complexity of fuzzy membership is $O(m)$. Hence, the computation of FULSTSVM is more than ULSTSVM, but the additional time is $O(m)$, which is insignificant w.r.t. cubic complexity of inverse calculation in FULSTSVM and ULSTSVM.

4 Experimental results

In this section, we perform numerical experiments, and show the comparative analysis of the results obtained on benchmark datasets. We also present two biomedical applications, viz. Alzheimer's disease and breast cancer to show the utility of the proposed FULSTSVM.

4.1 Data

All the real-world benchmark datasets are downloaded from the UCI [8], and KEEL repositories [1]. The MRI images used in this work are taken from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). ADNI was started in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The main objective 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. For breast cancer, the BreakHis histopathological dataset is utilized [26] in this work.

4.2 Setup and methodology

The experiments for all the algorithms are performed on a PC running on 64 bit Windows 10 operating system, with 2.30 GHz Intel® Xeon processor, and 128 GB of RAM with MATLAB R2017a environment. For cross validation, a 5-fold cross validation strategy is used for selecting the optimal parameters for all the algorithms. An additional optimization toolbox i.e. MOSEK optimization toolbox (<http://www.mosek.com>) is used for solving the QPPs of TWSVM and UTSVM.

For experiments on real-world datasets, the parameters are selected as follows: $c_1 = c_2 = c_u$, and $c_3 = c_4$ are selected from $\{10^{-5}, 10^{-4}, \dots, 10^5\}$, while μ is selected from the set $\{2^{-5}, 2^{-4}, \dots, 2^5\}$. The parameter ϵ is selected from $\{0.2, 0.4, 0.6, 0.8\}$. The universum is generated by averaging the samples randomly from the data [18, 21]. The training and testing data are chosen as 50% of total samples. For large scale datasets, we use fixed value of the hyper-parameters [24, 25]. Therefore, the value of $c_1 = c_2 = c_u$ is fixed as 10, and $c_3 = c_4$ is set as 10^{-5} , ϵ is selected as 0.7, and μ is chosen as 2 for all the algorithms. In all the algorithms, a radial basis function (RBF) kernel is used, which is defined as

$$K(p, q) = \exp\left(-\frac{\|p - q\|^2}{2\mu^2}\right), \quad (34)$$

where μ is a scalar parameter, and p and q are vectors.

In biomedical datasets, we used 150 structural MRI (T1) images from the ADNI database. The MRI images are preprocessed using Freesurfer pipeline [20, 31] to obtain the volumetric analysis of the brain. This resulted into volumetric data of 149 images, because 1 image failed to process. Therefore, the dataset includes 50 images each of control normal (CN), mild cognitive impairment (MCI),

Table 1 Comparative performance of proposed algorithm with existing approaches for classification on real-world benchmark datasets

Datasets (size)	TWSVM [13] Accuracy (c_1, μ) Time (s)	UTSVM [18] Accuracy (c_1, μ, ϵ) Time (s)	LSTSVM [15] Accuracy (c_1, μ) Time (s)	ULSTSVM [33] Accuracy (c_1, c_3, ϵ, μ) Time (s)	Proposed FULSTSVM Accuracy (c_1, c_3, ϵ, μ) Time (s)
Ecoli-0-1_vs_5 (240 × 12)	93.39 ($10^{-5}, 2^5$) 0.2566	95.04 ($10^{-4}, 2^4, 0.6$) 0.2598	93.39 ($10^{-5}, 2^5$) 0.0221	95.04 ($10^{-5}, 10^{-1}, 0.2, 2^4$) 0.0261	95.04 ($10^{-5}, 10^{-2}, 0.2, 2^4$) 0.0388
Ecoli-0-1-4-7_vs_5-6 (332 × 12)	95.81 ($10^{-4}, 2^5$) 0.0794	98.2 ($10^{-3}, 2^4, 0.4$) 0.0781	97.6 ($10^{-1}, 2^5$) 0.0321	98.8 ($10^0, 10^{-1}, 0.6, 2^5$) 0.0364	98.8 ($10^0, 10^{-2}, 0.6, 2^5$) 0.0444
Ecoli-0-2-3-4_vs_5 (202 × 7)	98.04 ($10^{-3}, 2^4$) 0.0881	98.04 ($10^{-4}, 2^5, 0.4$) 0.0898	97.06 ($10^1, 2^5$) 0.0118	94.12 ($10^{-2}, 10^{-5}, 0.4, 2^4$) 0.0129	99.02 ($10^{-2}, 10^{-5}, 0.6, 2^5$) 0.0161
Ecoli-0-2-6-7_vs_3-5 (224 × 7)	93.81 ($10^{-2}, 2^5$) 0.0332	94.69 ($10^{-2}, 2^5, 0.4$) 0.0379	97.35 ($10^2, 2^5$) 0.0141	92.92 ($10^{-1}, 10^{-5}, 0.6, 2^5$) 0.0157	94.69 ($10^0, 10^{-4}, 0.6, 2^5$) 0.019
Ecoli-0-4-6_vs_5 (202 × 6)	94.12 ($10^{-5}, 2^5$) 0.0274	94.12 ($10^{-4}, 2^5, 0.8$) 0.0384	94.12 ($10^3, 2^5$) 0.0155	95.1 ($10^0, 10^{-3}, 0.6, 2^5$) 0.018	96.08 ($10^0, 10^{-5}, 0.4, 2^5$) 0.0189
Ecoli-0-6-7_vs_3-5 (222 × 7)	91.07 ($10^{-4}, 2^4$) 0.0295	91.07 ($10^{-3}, 2^4, 0.4$) 0.0333	94.64 ($10^1, 2^5$) 0.0137	93.75 ($10^{-1}, 10^{-5}, 0.4, 2^5$) 0.0156	93.75 ($10^1, 10^{-3}, 0.2, 2^5$) 0.0177
Glass4 (214 × 9)	94.44 ($10^{-5}, 2^0$) 0.0341	95.37 ($10^{-1}, 2^3, 0.2$) 0.0304	94.44 ($10^{-5}, 2^0$) 0.0127	95.37 ($10^{-1}, 10^{-5}, 0.6, 2^1$) 0.0108	97.22 ($10^0, 10^{-4}, 0.6, 2^1$) 0.0192
Vehicle 1 (846 × 18)	74.06 ($10^{-4}, 2^5$) 0.2701	73.82 ($10^{-4}, 2^5, 0.8$) 0.3207	73.82 ($10^{-4}, 2^5$) 0.2116	72.88 ($10^{-4}, 10^{-5}, 0.6, 2^5$) 0.2364	73.82 ($10^{-4}, 10^{-5}, 0.2, 2^5$) 0.2319
Vehicle2 (846 × 18)	96.7 ($10^{-3}, 2^5$) 0.2824	98.11 ($10^{-3}, 2^5, 0.6$) 0.243	98.11 ($10^{-2}, 2^5$) 0.1469	98.11 ($10^0, 10^{-3}, 0.6, 2^5$) 0.2063	98.11 ($10^{-1}, 10^{-5}, 0.6, 2^5$) 0.2338
Pima Indians (768 × 8)	75.84 ($10^{-5}, 2^5$) 0.1783	76.1 ($10^{-5}, 2^5, 0.8$) 0.1807	74.81 ($10^{-5}, 2^5$) 0.1164	74.29 ($10^{-5}, 10^{-5}, 0.2, 2^5$) 0.1264	78.96 ($10^1, 10^1, 0.6, 2^5$) 0.1369
Yeast3 (1484 × 8)	94.35 ($10^{-1}, 2^0$) 0.6316	93.41 ($10^3, 2^2, 0.6$) 0.7756	93.67 ($10^{-2}, 2^0$) 0.463	93.81 ($10^{-2}, 10^{-5}, 0.2, 2^0$) 0.5962	94.21 ($10^3, 10^3, 0.8, 2^{-2}$) 0.495
Yeast1vs7 (458 × 8)	92.61 ($10^{-1}, 2^{-1}$) 0.0933	93.04 ($10^{-1}, 2^0, 0.4$) 0.079	92.61 ($10^0, 2^{-1}$) 0.0434	93.48 ($10^{-2}, 10^{-3}, 0.4, 2^{-2}$) 0.0471	93.48 ($10^{-2}, 10^{-3}, 0.4, 2^{-2}$) 0.0593
Ecoli0137vs26 (311 × 7)	97.44 ($10^{-1}, 2^3$) 0.0364	96.15 ($10^0, 2^0, 0.4$) 0.0428	94.87 ($10^{-3}, 2^{-2}$) 0.01876	95.51 ($10^{-2}, 10^{-1}, 0.2, 2^1$) 0.0206	96.15 ($10^{-5}, 10^0, 0.2, 2^{-1}$) 0.0244
Yeast5 (1484 × 8)	96.64 ($10^0, 2^4$) 0.805	96.64 ($10^{-4}, 2^{-2}, 0.2$) 0.7354	96.5 ($10^{-5}, 2^{-1}$) 0.4878	96.5 ($10^{-5}, 10^{-5}, 0.4, 2^{-1}$) 0.5705	96.77 ($10^{-4}, 10^{-3}, 0.6, 2^{-2}$) 0.5929

Table 1 (continued)

Datasets (size)	TWSVM [13] Accuracy (c_1, μ) Time (s)	UTSVM [18] Accuracy (c_1, μ, ϵ) Time (s)	LSTSVM [15] Accuracy (c_1, μ) Time (s)	ULSTSVM [33] Accuracy (c_1, c_3, ϵ, μ) Time (s)	Proposed FULSTSVM Accuracy (c_1, c_3, ϵ, μ) Time (s)
Cleveland (297×13)	81.21 ($10^{-1}, 2^4$) 0.0289	76.51 ($10^{-2}, 2^3, 0.8$) 0.0319	81.21 ($10^1, 2^4$) 0.0172	81.88 ($10^2, 10^3, 0.6, 2^2$) 0.0194	83.22 ($10^{-1}, 10^{-4}, 0.6, 2^5$) 0.0254
Transfusion (748×4)	78.93 ($10^{-3}, 2^5$) 0.2164	78.93 ($10^{-3}, 2^5, 0.2$) 0.2377	81.07 ($10^{-3}, 2^5$) 0.1115	82.67 ($10^2, 10^1, 0.4, 2^5$) 0.1207	81.87 ($10^1, 10^1, 0.4, 2^5$) 0.1559
Breast cancer wisconsin (682×9)	98.25 ($10^{-4}, 2^3$) 0.1431	98.25 ($10^{-4}, 2^3, 0.2$) 0.1971	98.54 ($10^0, 2^5$) 0.0855	98.25 ($10^0, 10^1, 0.8, 2^4$) 0.095	98.54 ($10^3, 10^2, 0.4, 2^5$) 0.1408
Ripley (1250×2)	90.58 ($10^0, 2^{-1}$) 0.5155	91.05 ($10^0, 2^{-1}, 0.4$) 0.5511	91.21 ($10^0, 2^{-1}$) 0.3014	91.05 ($10^3, 10^5, 0.8, 2^{-3}$) 0.3378	91.05 ($10^{-2}, 10^{-5}, 0.4, 2^{-1}$) 0.4951
Average rank	3.5556	3.2222	3.3056	3.1111	1.8056

Accuracy is in percentage, and average rank is calculated on accuracy. Bold values indicate highest accuracy

and 49 images of Alzheimer's disease (AD). A total of 91 features are extracted including 23 subcortical, 34 white matter volumes, and 34 cortical thickness values.

For breast cancer, the images are converted to gray level, and features are extracted using wavelet transform (Daubechies-4) up to 3 levels of decomposition. The approximation and detail coefficients are concatenated to form the feature vector [10]. A total of 314 breast cancer images include ADN- adenosis (benign), and DC- ductal carcinoma (cancer).

4.3 Real-world data

The results on 18 real-world benchmark datasets are presented in Table 1. For comparison, we have used TWSVM [13], UTSVM [18], LSTSVM [15], and ULSTSVM [33] algorithms. One can observe in Table 1 that the proposed FULSTSVM obtained the highest accuracies in 11 datasets. FULSTSVM outperformed the existing algorithms by obtaining an average rank of 1.8056 on accuracy values. This is due to the use of fuzzy memberships for all the data points in the proposed FULSTSVM. It is noticeable that the proposed FULSTSVM achieved the highest accuracy of 98.54% for Breast cancer wisconsin dataset with LSTSVM. However, ULSTSVM achieved a lesser accuracy of 98.25%, due to equal weighting to all the universal data points in ULSTSVM.

One can observe in Table 1 that the training time of proposed FULSTSVM is lower than TWSVM and UTSVM algorithms. This is because TWSVM and UTSVM solve a

pair of QPPs, which is computationally expensive. On the other hand proposed FULSTSVM solves a system of linear equations. However, the training time of FULSTSVM is higher than LSTSVM, and ULSTSVM because of the fuzzy membership calculations.

4.3.1 Statistical significance

In order to prove the statistical significance of the proposed FULSTSVM for generalization performance, we perform the Friedman and Nemenyi post hoc test [7].

The Friedman test is performed using the average ranks of the algorithms from Table 1. Here, we first assume that all the algorithms are not significantly different, as the null hypothesis. Then, we calculate the χ_F^2 value as

$$\chi_F^2 = \frac{12N}{p(p+1)} \left[\sum_{i=1}^p R_i^2 - \frac{p(p+1)^2}{4} \right], \quad (35)$$

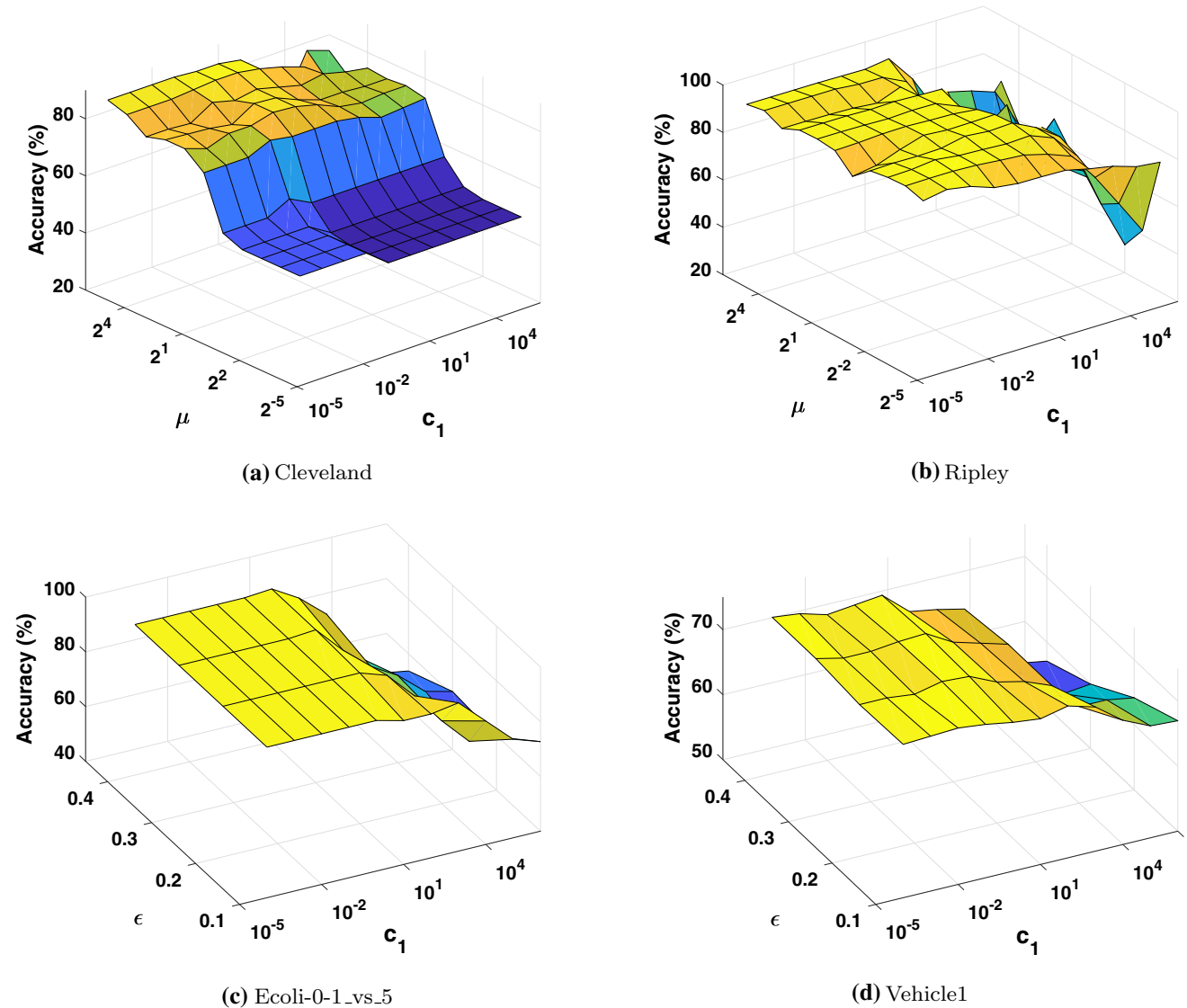
where N is number of datasets, p is the number of algorithms, and R_i is average rank for the methods.

$$\begin{aligned} \chi_F^2 &= \frac{12 \times 18}{5(5+1)} \\ &\quad \left[(3.5556^2 + 3.2222^2 + 3.3056^2 + 3.1111^2 + 1.8056^2) - \frac{5(5+1)^2}{4} \right], \\ &\approx 13.6151. \end{aligned} \quad (36)$$

The F_F value is obtained as

Table 2 Significant difference between the proposed FULSTSVM and existing algorithms in pairwise comparison

Significance	TWSVM	UTSVM	LSTSVM	ULSTSVM
Proposed FULSTSVM	Yes	Yes	Yes	Yes

**Fig. 2** Insensitivity analysis of proposed FULSTSVM for c_1 and μ in (a) and (b), and for c_1 and ϵ in (c) and (d) on real-world benchmark datasets

$$F_F = \frac{(18-1)(13.6151)}{18 \times (5-1) - 13.6151} \approx 3.9643. \quad (37)$$

In this case, the F -distribution has $(5-1, (5-1)(18-1)) = (4, 68)$ degrees of freedom. Therefore, the critical value for $F(6, 150)$ at $\alpha = 0.05$ level of significance is 2.5066. Since, $F_F = 3.9643 > 2.5066$, we reject the null hypothesis. Thus, there is significant difference between these methods.

Next, for pairwise difference, we use the Nemenyi posthoc test [7] to check pairwise difference between proposed FULSTSVM and existing algorithms. The critical difference for our case at $\alpha = 0.10$ level of significance level is $2.459 \sqrt{\frac{5(5+1)}{6 \times 18}} \approx 1.296$. The pairwise difference of the average ranks should be greater than CD for significance. Table 2 shows the pairwise significant difference between the methods based on average ranks. One can observe that proposed FULSTSVM is significantly

Table 3 Comparative performance of the proposed and baseline algorithms on Alzheimer's and breast cancer datasets

Datasets	TWSVM [13] Accuracy (c_1, μ) Time (s)	UTSVM [18] Accuracy (c_1, μ, ϵ) Time (s)	LSTSVM [15] Accuracy (c_1, μ) Time (s)	ULSTSVM [33] Accuracy (c_1, c_3, ϵ, μ) Time (s)	Proposed FULSTSVM Accuracy (c_1, c_3, ϵ, μ) Time (s)
CN_AD	80 ($10^{-1}, 2^5$) 0.255	80 ($10^{-1}, 2^5, 0.2$) 0.2427	77.5 ($10^{-5}, 2^1$) 0.0105	75 ($10^1, 10^4, 0.8, 2^4$) 0.0149	72.5 ($10^{-5}, 10^4, 0.2, 2^4$) 0.0255
CN_MCI	69.23 ($10^{-5}, 2^5$) 0.0311	66.67 ($10^{-2}, 2^2, 0.8$) 0.0328	69.23 ($10^{-5}, 2^5$) 0.0062	71.79 ($10^{-1}, 10^{-4}, 0.2, 2^5$) 0.0081	69.23 ($10^0, 10^{-3}, 0.2, 2^4$) 0.0109
AD_MCI	53.85 ($10^{-5}, 2^5$) 0.094	58.97 ($10^{-1}, 2^0, 0.2$) 0.0677	48.72 ($10^{-1}, 2^5$) 0.006	48.72 ($10^{-2}, 10^{-5}, 0.8, 2^{-1}$) 0.0054	66.67 ($10^{-5}, 10^{-4}, 0.6, 2^5$) 0.0094
ADN_DC	83.54 ($10^{-5}, 2^3$) 0.2538	83.54 ($10^{-4}, 2^4, 0.4$) 0.2968	83.54 ($10^{-5}, 2^3$) 0.0732	82.28 ($10^{-5}, 10^0, 0.2, 2^4$) 0.0939	84.18 ($10^{-5}, 10^{-1}, 0.2, 2^5$) 0.1072
Average rank	2.625	2.875	3.375	3.625	2.5

Accuracy is in percentage. Bold values indicate highest accuracy

different from TWSVM, UTSVM, LSTSVM, and ULSTSVM algorithms.

4.4 Insensitivity analysis

To check the effect of hyper-parameter values on the accuracy of the proposed FULSTSVM, we present the insensitivity analysis. The insensitivity performance is checked for the penalty parameter c_1 , with kernel parameter μ , and c_1 with ϵ parameter of the insensitive zone. The variation of accuracy of FULSTSVM for these parameters is shown for four datasets in Fig. 2.

Figure 2a, b shows the change in accuracy for different values of c_1 and μ . One can observe that accuracy of proposed FULSTSVM is higher for lower values of c_1 , and higher values of μ . The variation in accuracy for c_1 with ϵ is shown in Fig. 2c, d. The parameter ϵ is not affecting the accuracy in a significant manner. However, here also the accuracy of FULSTSVM is higher for lesser values of the

hyper-parameter c_1 . This also justifies the parameter selection in the experiments.

4.5 Biomedical data

In this section, we present the results on classification of Alzheimer's disease and breast cancer datasets. The results for these applications are shown in Table 3. One can observe that the proposed FULSTSVM performs better than baseline algorithms in most of the cases. This is reflected in the average rank based on accuracy. Proposed FULSTSVM obtained lowest average rank of 2.5. The accuracy of FULSTSVM is higher than other algorithms for AD_MCI, which is a difficult classification problem [28].

Moreover, for breast cancer data i.e. adenosis vs ductal carcinoma, proposed FULSTSVM obtains highest accuracy of 84.18%. The better average rank of proposed FULSTSVM for accuracy can be attributed to the use of

Table 4 Classification performance of the proposed and baseline algorithms on large-scale datasets

Datasets	LSTSVM [15] Accuracy time (s)	ULSTSVM[33] Accuracy time (s)	Proposed FULSTSVM Accuracy time (s)
Skin-5k	93.56	97.8	97.76
	4.9931	5.7513	6.1476
Skin-10k	95.26	97.64	97.88
	20.8591	24.0887	24.4975
Skin-20k	97.03	98.8	98.91
	97.2516	104.539	105.941
Skin-30k	97.74	99.23	99.25
	233.092	243.475	247.076
Skin-40k	98.6	99.47	99.53
	427.891	453.43	548.742

Accuracy is in percentage. Bold values indicate highest accuracy

fuzzy membership with universum data. It leads to prior information for the model, with less sensitivity to outlier data points in the classes, as well as in the universum. This implies the applicability of the proposed FULSTSVM for biomedical applications.

4.6 Large-scale data

In order to check the performance of our proposed FULSTSVM on large datasets, we used the Skin segmentation dataset from UCI repository [8]. For comparison, we used two other efficient algorithms, viz. LSTSVM, and ULSTSVM. The results are shown in Table 4. It is observable that the proposed FULSTSVM is showing higher accuracy on most of the datasets. This is because FULSTSVM removes the effect of outliers in the generation of universum data, whereas ULSTSVM gives equal importance to all the universum data points. Moreover, the proposed FULSTSVM also gives proper weighting to the data points of the binary classes, where LSTSVM gives equal to weights to all the data points. However, the time is the least in case of LSTSVM, because there is no universum data in LSTSVM. The time is slightly higher in FULSTSVM as compared to ULSTSVM due to the calculation of fuzzy membership values. However, the additional time in FULSTSVM is not significant as discussed in Sect. 3.4 in terms of time complexity.

5 Conclusions and future work

In this work, to deal with noisy datasets, we proposed a novel and efficient fuzzy based learning algorithm, termed as fuzzy universum least squares twin support vector machine (FULSTSVM). The proposed algorithm gives prior information about data distribution to the classifier, and also provides fuzzy membership to the data points and universum. Moreover, the optimization problem of FULSTSVM is solved by a system of linear equations. This makes FULSTSVM efficient in terms of training time. The proposed FULSTSVM is a robust universum based algorithm for classification of data with outliers. Statistical tests on experimental results confirm the significance of the proposed algorithm. Proposed FULSTSVM also performed better on large sized datasets in terms of accuracy, showing its scalability on large datasets.

Results on applications i.e. Alzheimer's disease and breast cancer clearly show the applicability of the proposed FULSTSVM for healthcare data. In future, the proposed FULSTSVM can be improved by implementing new techniques for selecting the universum. The universum data can be selected from a dataset related to a particular application. Moreover, novel fuzzy membership functions

can be used with the proposed FULSTSVM in various applications. Proposed FULSTSVM can also be extended for class imbalanced and multiclass problems. The code of proposed FULSTSVM will be available on the author's homepage: <https://github.com/mtanveer1/>.

Acknowledgements The funding for this work is obtained from Science and Engineering Research Board (SERB), INDIA under Ramanujan fellowship grant no. SB/S2/RJN-001/2016, and also under Early Career Research Award grant no. ECR/2017/000053. We also acknowledge Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA for funding under Extra Mural Research (EMR) Scheme grant no. 22(0751)/17/EMR-II. We want to acknowledge our institute, the Indian Institute of Technology Indore for providing various facilities and resources for this work. We also thank 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.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J Multiple-Valued Logic Soft Comput* 17:255–287
2. Bai L, Shao YH, Wang Z, Li CN (2019) Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding. *Knowl-Based Syst* 163:227–240

3. Balasundaram S, Tanveer M (2012) On proximal bilateral-weighted fuzzy support vector machine classifiers. *Int J Adv Intell Paradigms* 4(3–4):199–210
4. Batuwita R, Palade V (2010) Fsvm-cil: fuzzy support vector machines for class imbalance learning. *IEEE Trans Fuzzy Syst* 18(3):558–571
5. Cervantes J, Garcia-Lamont F, Rodriguez-Mazahua L, Lopez A (2020) A comprehensive survey on support vector machine classification: applications, challenges and trends. *Neurocomputing* 408:189–215
6. Cherkassky V, Dhar S, Dai W (2011) Practical conditions for effectiveness of the universum learning. *IEEE Trans Neural Netw* 22(8):1241–1255
7. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
8. Dua D, Graff C (2014) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
9. Frozza RL, Lourenco MV, De Felice FG (2018) Challenges for Alzheimer's disease therapy: insights from novel mechanisms beyond memory defects. *Front Neurosci* 12:37
10. Gautam C, Mishra PK, Tiwari A, Richhariya B, Pandey HM, Wang S, Tanveer M (2020) ADNI: minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data. *Neural Netw* 123:191–216
11. Hao PY, Kung CF, Chang CY, Ou JB (2020) Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane. *Appl Soft Comput* 98:106806
12. Huang X, Guo F (2020) A kernel fuzzy twin SVM model for early warning systems of extreme financial risks. *Int J Financ Econ*. <https://doi.org/10.1002/ijfe.1858>
13. Jayadeva C, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. *IEEE Trans Pattern Anal Mach Intell* 29(5):905–910
14. Khemchandani R, Jayadeva CS (2009) Regularized least squares fuzzy support vector regression for financial time series forecasting. *Expert Syst Appl* 36(1):132–138
15. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. *Expert Syst Appl* 36(4):7535–7543
16. Lin CF, Wang SD (2002) Fuzzy support vector machines. *IEEE Trans Neural Netw* 13(2):464–471
17. Mello AR, Stemmer MR, Koerich AL (2020) Incremental and decremental fuzzy bounded twin support vector machine. *Inf Sci* 526:20–38
18. Qi Z, Tian Y, Shi Y (2012) Twin support vector machine with universum data. *Neural Netw* 36:112–119
19. Qin G, Lu X (2018) Integration of weighted LS-SVM and manifold learning for fuzzy modeling. *Neurocomputing* 282:184–191
20. Reuter M, Schmansky NJ, Rosas HD, Fischl B (2012) Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage* 61(4):1402–1418
21. Richhariya B, Tanveer M (2018) EEG signal classification using universum support vector machine. *Expert Syst Appl* 106:169–182
22. Richhariya B, Tanveer M (2018) A robust fuzzy least squares twin support vector machine for class imbalance learning. *Appl Soft Comput* 71:418–432
23. Richhariya B, Tanveer M (2019) A fuzzy universum support vector machine based on information entropy. In: Tanveer M, Pachori RB (eds) *Machine Intelligence and Signal Analysis. Advances in Intelligent Systems and Computing*. Springer, Singapore, pp 569–582. https://doi.org/10.1007/978-981-13-0923-6_49
24. Richhariya B, Tanveer M (2020) Alzheimer's disease neuroimaging initiative: an efficient angle based universum least squares twin support vector machine for pattern classification. *ACM Trans Internet Technol*. <https://doi.org/10.1145/3387131>
25. Shao YH, Deng NY, Yang ZM (2012) Least squares recursive projection twin support vector machine for classification. *Pattern Recogn* 45(6):2299–2307
26. Spanhol F, Oliveira L, Petitjean C, Heutte L (2015) A dataset for breast cancer histopathological image classification. *IEEE Trans Biomed Eng* 63(7):1455–1462
27. Tanveer M, Khan MA, Ho SS (2016) Robust energy-based least squares twin support vector machines. *Appl Intell* 45(1):174–186
28. Tanveer M, Richhariya B, Khan RU, Rashid AH, Khanna P, Prasad M, Lin CT (2020) Machine learning techniques for the diagnosis of Alzheimer's disease: a review. *ACM Trans Multimed Comput Commun Appl* 16(1):1–35
29. Tomar D, Agarwal S (2015) Hybrid feature selection based weighted least squares twin support vector machine approach for diagnosing breast cancer, hepatitis, and diabetes. *Adv Artif Neural Syst* 2015:265637. <https://doi.org/10.1155/2015/265637>
30. Wang TY, Chiang HM (2007) Fuzzy support vector machine for multi-class text categorization. *Inf Process Manag* 43(4):914–929
31. Westman E, Muehlboeck JS, Simmons A (2012) Combining MRI and CSF measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion. *NeuroImage* 62(1):229–238
32. Weston J, Collobert R, Sinz F, Bottou L, Vapnik V (2006) Inference with the universum. In: *Proceedings of the 23rd international conference on machine learning*. ACM, pp 1009–1016
33. Xu Y, Chen M, Li G (2016) Least squares twin support vector machine with universum data for classification. *Int J Syst Sci* 47(15):3637–3645
34. Yue W, Wang Z, Chen H, Payne A, Liu X (2018) Machine learning with applications in breast cancer diagnosis and prognosis. *Designs* 2(2):13
35. Zhang T, Chen W, Li M (2019) Classification of inter-ictal and ictal EEGs using multi-basis MODWPT, dimensionality reduction algorithms and LS-SVM: a comparative study. *Biomed Signal Process Control* 47:240–251
36. Zhou X, Jiang W, Tian Y, Shi Y (2010) Kernel subclass convex hull sample selection method for SVM on face recognition. *Neurocomputing* 73(10–12):2234–2246

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.