Neural Comput & Applic (2018) 29:553–563
https://doi.org/10.1007/s00521-016-2468-4

ORIGINAL ARTICLE

Fuzzy least squares twin support vector clustering

Reshma Khemchandani[1] · **Aman Pal**[1] · **Suresh Chandra**[2]

Received: 12 September 2015 / Accepted: 6 July 2016 / Published online: 16 July 2016

Abstract In this paper, we have formulated a fuzzy least squares version of recently proposed clustering method, namely twin support vector clustering (TWSVC). Here, a fuzzy membership value of each data pattern to different cluster is optimized and is further used for assigning each data pattern to one or other cluster. The formulation leads to finding k cluster center planes by solving modified primal problem of TWSVC, instead of the dual problem usually solved. We show that the solution of the proposed algorithm reduces to solving a series of system of linear equations as opposed to solving series of quadratic programming problems along with system of linear equations as in TWSVC. The experimental results on several publicly available datasets show that the proposed fuzzy least squares twin support vector clustering (F-LS-TWSVC) achieves comparable clustering accuracy to that of TWSVC with comparatively lesser computational time. Further, we have given an application of F-LS-TWSVC for segmentation of color images.

Keywords Machine learning · Twin support vector clustering · Plane-based clustering · Fuzzy clustering

✉ Reshma Khemchandani
reshma.khemchandani@sau.ac.in

Aman Pal
aman.pal@students.sau.ac.in

Suresh Chandra
chandra@maths.iitd.ac.in

[1] South Asian University, New Delhi, India

[2] Indian Institute of Technology, New Delhi, India

1 Introduction

Clustering is a powerful tool which aims at grouping similar objects into the same cluster and dissimilar objects into different clusters by identifying dominant structures in the data. It has remained a widely studied research area in machine learning [1, 2] and has applications in diverse domains such as computer vision, text mining, bioinformatics and signal processing [3–6].

Traditional point-based clustering methods such as k-means [1] and k-median [7] work by partitioning the data into clusters based on the cluster prototype points. These methods perform poorly in case when data are not distributed around several cluster points. In contrast to these, plane-based clustering methods such as k-plane clustering [8], proximal plane clustering [9] and local k-proximal plane clustering [10] have been proposed in the literature. These methods calculate k cluster center planes and partition the data into k clusters according to the proximity of the data points with these k planes.

Jayadeva et al. [11] have proposed twin support vector machine (TWSVM) classifier for binary data classification where the two hyperplanes are obtained by solving two related smaller-sized quadratic programming problems (QPPs) as compare to single large-sized QPP in conventional support vector machine (SVM). Mehrkanoon et al. [12] introduced a general framework of non-parallel support vector machines, which involves a regularization term, a scatter loss and a misclassification loss. Taking motivation from Xie and Sun [11, 13], they have proposed multi-view twin support vector machines in the semi-supervised learning framework which combines two views by introducing the constraint of similarity between two one-dimensional projections identifying two distinct TWSVMs from two feature spaces. An inherent shortcoming of twin

support vector machines is that the resultant hyperplanes are very sensitive to outliers in data. To overcome this disadvantage, Xie and Sun [14] have proposed multitask centroid twin support vector machines. The more recent extensions and developments in TWSVMs have been discussed in [15, 16].

Recently, Shao et al. [17] proposed a novel plane-based clustering method, namely twin support vector clustering (TWSVC). The method is based on twin support vector machine (TWSVM) [11] and exploits information from both within and between clusters. Different from the TWSVM, the formulation of TWSVC is modified to get one cluster plane close to the points of its own cluster and at the same time far away from the points of different clusters from both sides of cluster plane. Experimental results in [17] show the superiority of the method against existing plane-based methods.

Working on the lines of [18], in this paper, we first extend the TWSVC to least squares TWSVC (LS-TWSVC) and further propose a fuzzy extension of LS-TWSVC termed as F-LS-TWSVC by incorporating a fuzzy matrix which represents the membership value of each data point to different available clusters. The key features of F-LS-TWSVC are listed below:

- We modify the quadratic programming problem (QPP)-based formulation of TWSVC in least squares sense which leads to solving optimization problem with equality constraints.
- A regularization term in the objective function is introduced which takes care of structural risk component along with empirical risk associated with data samples.
- The solution of LS-TWSVC requires solving series of system of linear equations as opposed to solving a series of QPP and a system of linear equations as in the case of TWSVC.
- We incorporate fuzzy membership matrix of each data sample to different clusters in order to extend LS-TWSVC to F-LS-TWSVC. The initial fuzzy membership matrix is obtained using fuzzy nearest neighbor algorithm [19] (as discussed in Sect. 5.3).
- Experimental results on several benchmark UCI datasets indicate that the proposed F-LS-TWSVC achieves similar or better clustering accuracy results as compared to TWSVC and with considerably lesser computational time for both linear as well as nonlinear cases.
- We also perform experiments on image segmentation as an application to our proposed formulation.

The paper is organized as follows. In Sect. 2, we briefly discuss k-means and TWSVC. Section 3 presents the formulation of LS-TWSVC and F-LS-TWSVC along with algorithm in detail. Section 4 discusses the nonlinear extension of LS-TWSVC and F-LS-TWSVC, respectively. Computational comparison of proposed formulation with other plane-based formulations is done in Sect. 5. Section 6 provides the concluding remarks.

2 Background and related work

The samples are denoted by a set of m row vectors $X = \{x_1; x_2; \ldots; x_m\}$ in the n-dimensional real space $\mathbb{R}^n$, where the j *th* sample is $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$. We assume that these samples belong to k clusters with their corresponding cluster labels in $\{1, 2, \ldots, k\}$. Let X_i denotes the set of samples belonging to cluster label i and $\overline{X_i}$ denotes the set of samples belonging to remaining cluster labels, where $i = 1, 2, \ldots, k$. The fuzzy membership of a sample is denoted by k column vector $\{s_1, s_2, \ldots, s_k\}$ where s_j, $j = 1, .., k$ represents the fuzzy membership value of all samples in the j *th* cluster. Let S_i and $\overline{S_i}$ denote the diagonal fuzzy membership matrix corresponding to samples belonging to cluster label i and remaining cluster labels, respectively, where $i = 1, 2, \ldots, k$ whose diagonal entries represent the association of i *th* pattern to j *th* cluster.

2.1 k-Means

Consider the clustering problem with a set X of m unlabeled data samples in $\mathbb{R}^n$. k-means [1] partition X into k clusters $X_1, X_2, \ldots, X_k$ such that the data samples are close to their respective k cluster center points $\mu_1, \mu_2, \ldots, \mu_k$. It aims to minimize the following objective function

$$\min_{(\mu_1, \mu_2, \ldots, \mu_k, X_1, X_2, \ldots, X_k)} \quad \sum_{i=1}^{k} \sum_{j=1}^{m_i} ||X_i(j) - \mu_i||_2, \tag{1}$$

where $X_i(j)$ represents the j *th* sample in X_i, m_i is the number of samples in X_i so that $m_1 + m_2 + \cdots + m_k = m$, and $||.||_2$ denotes L_2 norm.

In practice, an iterative relocation algorithm is followed which minimize (1) locally. Given an initial set of k cluster center points, each sample x is labeled to its nearest cluster center by

$$y = \arg_i \quad \min\{||x - \mu_i||, \quad i = 1, 2, \ldots, k\}. \tag{2}$$

Then the k cluster center points are updated as the mean of the corresponding cluster samples since for a given assignment X_i, the mean of the cluster samples represents the solution to (1). At each iteration, the cluster centers and sample labels are updated until some convergence criteria is satisfied.

2.2 TWSVC

Working on the lines of TWSVM, Wang et al. [17] proposed TWSVC. In TWSVC, the following problem has been considered in order to obtain k cluster center planes $w_i^{\mathrm{T}}x + b_i = 0$, $i = 1, 2, \ldots, k$, one for each cluster:

$$\begin{aligned} \underset{(w_i,\, b_i,\, q_i, X_i)}{\mathrm{Min}} \; & \frac{1}{2}\|(X_i w_i + b_i e)\|_2^2 + Ce^{\mathrm{T}} q_i \\ \text{s.t.} \; & |(\overline{X_i} w_i + b_i e)| + q_i \geq e, \\ & q_i \geq 0, \end{aligned} \tag{3}$$

where $C > 0$ is a penalty parameter and q_i is a slack vector corresponding to *i th* cluster. Here, $|\cdot|$ would illustrate the condition that the *i th* cluster center plane is required to be close to the pattern of cluster X_i and away from the other cluster $\overline{X_i}$ from both sides.

Each of the k hyperplane is close to the samples of its own cluster and far away from the samples of the other clusters from both sides unlike the One Against All (OAA)-based multi-class TWSVM which yields hyperplanes which are close to the samples of its cluster but are away from the samples of other cluster from one side only.

For given a certain X_i Wang et al. [17] solved (3) by the concave–convex procedure (CCCP) [20], which decomposes it into a series of convex quadratic subproblems with an initial w_i^0 and b_i^0 as follows:

$$\begin{aligned} \underset{(w_i^{j+1},\, b_i^{j+1},\, q_i^{j+1})}{\mathrm{Min}} \; & \frac{1}{2}\|(X_i w_i^{j+1} + b_i^{j+1} e)\|_2^2 + Ce^{\mathrm{T}} q_i^{j+1} \\ \text{s.t.} \; & T(|(\overline{X_i} w_i^{j+1} + b_i^{j+1} e)|) + q_i^{j+1} \geq e, \\ & q_i^{j+1} \geq 0, \end{aligned} \tag{4}$$

where the index of the subproblem $j = 0, 1, 2, \ldots$, and $T(.)$ denotes the first-order Taylor expansion.

Wang et al. [17] showed that the above problem (4) is equivalent to the following optimization problem:

$$\begin{aligned} \underset{(w_i^{j+1},\, b_i^{j+1},\, q_i^{j+1})}{\mathrm{Min}} \; & \frac{1}{2}\|(X_i w_i^{j+1} + b_i^{j+1} e)\|_2^2 + Ce^{\mathrm{T}} q_i^{j+1} \\ \text{s.t.} \; & \mathrm{diag}(\mathrm{sign}(\overline{X_i} w_i^{j} + b_i^{j} e))(\overline{X_i} w_i^{j+1} + b_i^{j+1} e) + q_i^{j+1} \geq e, \\ & q_i^{j+1} \geq 0. \end{aligned} \tag{5}$$

The solution of (5) is obtained by solving its dual problem

$$\begin{aligned} \underset{\alpha}{\mathrm{Min}} \; & \frac{1}{2}\alpha^{\mathrm{T}} G (H^{\mathrm{T}} H)^{-1} G^{\mathrm{T}} \alpha - e^{\mathrm{T}} \alpha \\ \text{s.t.} \; & 0 \leq \alpha \leq Ce, \end{aligned} \tag{6}$$

where $G = \mathrm{diag}(\mathrm{sign}(\overline{X_i} w_i^{j} + b_i^{j} e))[\overline{X_i} \quad e]$, $H = [X_i \quad e]$ and $\alpha \in \mathbb{R}^{m - m_i}$ is the Lagrangian multiplier vector.

Once the solution of (6) is obtained, the decision variable $[w_i^{j+1}; b_i^{j+1}]$ is obtain from solving systems of linear equation

$$[w_i^{j+1}; b_i^{j+1}]^{\mathrm{T}} = (H^{\mathrm{T}} H)^{-1} G^{\mathrm{T}} \alpha. \tag{7}$$

In short, for each $i = 1, 2, \ldots, k$, we select an initial w_i^0 and b_i^0 and solve for $[w_i^{j+1}; b_i^{j+1}]$ by (7) for $j = 0, 1, 2\ldots$, and stop when $\|[w_i^{j+1}; b_i^{j+1}] - [w_i^{j}; b_i^{j}]\|$ is small enough. We then set $w_i = w_i^{j+1}$, $b_i = b_i^{j+1}$.

Given any initial sample cluster assignment of X, TWSVC iterates alternatively updating the cluster center planes by solving (3) with a certain X_i and then updating cluster assignments by relabeling each sample by $y = \arg\min_i \{|w_i^{\mathrm{T}} x + b_i|, i = 1, 2, \ldots, k\}$. The iterations are repeated until some convergence criteria is met.

It is to be noted that the solution of (5) requires solving a QPP with $m - m_i$ parameters and in addition requires an inversion of matrix of size $(n+1) \times (n+1)$ where $n << m$.

TWSVC was also extended in [17] to handle nonlinear case by considering k cluster center kernel-generated surfaces for $i = 1, 2, \ldots, k$

$$K(x, X) u_i + \gamma_i = 0, \tag{8}$$

where K is any arbitrary kernel, $u_i \in \mathbb{R}^m$ and $\gamma \in \mathbb{R}$. The kernel counterpart of (3) for $i = 1, 2, \ldots, k$ is

$$\begin{aligned} \underset{(u_i,\, \gamma_i,\, \eta_i, X_i)}{\mathrm{Min}} \; & \frac{1}{2}\|(K(X_i, X) u_i + \gamma_i e)\|_2^2 + Ce^{T} \eta_i \\ \text{s.t.} \; & |(K(\overline{X_i}, X) u_i + \gamma_i e)| + \eta_i \geq e, \\ & \eta_i \geq 0, \end{aligned} \tag{9}$$

where η_i is a slack vector. The above problem is solved in a similar manner to linear case by CCCP. However, it is worth mentioning that for each i $(i = 1, 2, \ldots, k)$ the solution of nonlinear TWSVC is decomposed into solving a series of subproblems which requires inversion of matrix of size $(m+1) \times (m+1)$ along with a QPP to be solved, where m is the total number of patterns.

3 Fuzzy least squares twin support vector clustering

Taking motivation from [21], we first propose least squares version of TWSVC and then extend it to fuzzy LS-TWSVC.

Here, we modify the primal problem of linear TWSVC (3) in least squares sense, with inequality constraints replaced by equality constraints along with adding a regularization term in the objective function to incorporate structural risk minimization (SRM) principle. Thus, for cluster i $(i = 1, 2, \ldots, k)$ the optimization problem is given as:

$$\underset{(w_i,\, b_i,\, q_i, X_i)}{\text{Min}} \frac{1}{2}\|(X_i w_i + b_i e)\|_2^2 + \frac{v}{2}\left(\|w_i\|_2^2 + b_i^2\right) + \frac{C}{2}\|q_i\|_2^2$$
$$\text{s.t.} \quad |(\overline{X_i} w_i + b_i e)| + q_i = e, \tag{10}$$

where $v > 0$ is a parameter. Note that QPP (10) uses the square of L_2-norm of slack variable q_i instead of L_1-norm of q_i in (3), which makes the constraint $q_i \geq 0$ redundant [18]. Solving (10) is equivalent to solving system of linear equations.

Further, we introduce the fuzzy matrices S_i and $\overline{S_i}$ in (10) which indicates the fuzzy membership value of each data points to different available clusters as follows:

$$\underset{(w_i,\, b_i,\, q_i, X_i)}{\text{Min}} \frac{1}{2}\|((S_i X_i) w_i + b_i e)\|_2^2 + \frac{v}{2}\left(\|w_i\|_2^2 + b_i^2\right) + \frac{C}{2}\|q_i\|_2^2$$
$$\text{s.t.} \quad |((\overline{S_i X_i}) w_i + b_i e)| + q_i = e. \tag{11}$$

Similar to the solution of TWSVC formulation [17], the above optimization problem can be solved by using the concave–convex procedure (CCCP) [20], which decomposes it into a series of j $(j = 1, 2, \ldots)$ quadratic subproblems with initial w_i^0 and b_i^0 as follows:

$$\underset{\left(w_i^{j+1},\, b_i^{j+1},\, q_i^{j+1}\right)}{\text{Min}} \frac{1}{2}\|\left((S_i X_i) w_i^{j+1} + b_i^{j+1} e\right)\|_2^2 + \frac{v}{2}\left(\|w_i^{j+1}\|_2^2 + (b_i^{j+1})^2\right) + \frac{C}{2}\|q_i^{j+1}\|_2^2 \tag{12}$$
$$\text{s.t.} \quad T\left(|((\overline{S_i X_i}) w_i^{j+1} + b_i^{j+1} e)|\right) + q_i^{j+1} = e,$$

where $T(.)$ denotes the first-order Taylor expansion.

Working along the lines of [17], the equation (12) reduces to

$$\underset{\left(w_i^{j+1},\, b_i^{j+1},\, q_i^{j+1}\right)}{\text{Min}} \frac{1}{2}\|\left((S_i X_i) w_i^{j+1} + b_i^{j+1} e\right)\|_2^2 + \frac{v}{2}\left(\|w_i^{j+1}\|_2^2 + (b_i^{j+1})^2\right) + \frac{C}{2}\|q_i^{j+1}\|_2^2$$
$$\text{s.t.} \quad \text{diag}\left(\text{sign}((\overline{S_i X_i}) w_i^j + b_i^j e)\right)\left((\overline{S_i X_i}) w_i^{j+1} + b_i^{j+1} e\right) + q_i^{j+1} = e. \tag{13}$$

Substituting the error variable q_i^{j+1} into the objective function of (13) leads to the following optimization problem.

$$\underset{\left(w_i^{j+1},\, b_i^{j+1}\right)}{\text{Min}} \frac{1}{2}\|\left((S_i X_i) w_i^{j+1} + b_i^{j+1} e\right)\|_2^2 + \frac{v}{2}\left(\|w_i^{j+1}\|_2^2 + (b_i^{j+1})^2\right) + \frac{C}{2}\|\text{diag}\left(\text{sign}((\overline{S_i X_i}) w_i^j + b_i^j e)\right)\left((\overline{S_i X_i}) w_i^{j+1} + b_i^{j+1} e\right) - e\|_2^2. \tag{14}$$

Further, considering the gradient of (14) with respect to w_i^{j+1} and b_i^{j+1} and equate it to zero gives:

$$(S_i X_i)^{\mathrm{T}}\left[H_1 z_i^{j+1}\right] + v w_i^{j+1} + C(\overline{S_i X_i})^{\mathrm{T}} G^{\mathrm{T}}\left[G(H_2 z_i^{j+1}) - e\right] = 0, \tag{15}$$

$$e^{\mathrm{T}}\left[H_1 z_i^{j+1}\right] + v b_i^{j+1} + C e^{\mathrm{T}} G^{\mathrm{T}}\left[G(H_2 z_i^{j+1}) - e\right] = 0, \tag{16}$$

where $H_1 = [S_i X_i \quad e]$, $H_2 = [\overline{S_i X_i} \quad e]$, $z_i^{j+1} = [w_i^{j+1}; b_i^{j+1}]$ and $G = \text{diag}(\text{sign}(H_2\ z_i^j))$. Rearranging the above equations, we obtained the following system of linear equations:

$$\left(H_1^{\mathrm{T}} H_1 + Iv + C H_2^{\mathrm{T}} H_2\right) z_i^{j+1} = C H_2^{\mathrm{T}} G^T e, \tag{17}$$

which gives the solution for z_i^{j+1}:

$$z_i^{j+1} = \left[w_i^{j+1}; b_i^{j+1}\right] = C(H_1^{\mathrm{T}} H_1 + Iv + C H_2^{\mathrm{T}} H_2)^{-1} H_2^{\mathrm{T}} G^T e. \tag{18}$$

Input : The dataset X; the number of clusters k; appropriate F-LS-TWSVC parameters C, ν.
Output: k fuzzy matrices S^i for $i = 1, 2, \ldots, k$
Process:
1. Initialize fuzzy membership matrix S via FNNG(as explained in 5.3) for each data points in k clusters.
2. For each $i = 1, 2, \ldots, k$:
   2.1. Use obtained fuzzy membership matrix in Step 1 as initial fuzzy membership matrix S_0^j and solve equ.(18) to obtain $[w_i^{j+1} \quad b_i^{j+1}]$, $j = 0, 1, 2\ldots.$
   $S_i^{j+1} = \frac{1}{d^{j+1}}$
   2.2. Stop when $\|S_i^{j+1} - S_i^j\| < \epsilon$ and set $S_i = S_i^{j+1}$
3. Update the cluster assignments by relabelling each sample by $y = \underset{i}{arg\ max}\{S_i\}$.

Algorithm 1: F-LS-TWSVC clustering algorithm

It can be finally observed that our algorithm requires the solution of (18) which involves inversion of smaller-dimensional matrix of size $(n+1)\times(n+1)$ as compared to an additional QPP solution required in case of TWSVC. The details of the proposed algorithm are described in Algorithm 1.

4 Nonlinear fuzzy least squares twin support vector clustering

Working on the lines of [11], we extend the nonlinear formulation of F-LS-TWSVC by considering k cluster center kernel-generated surfaces for $i = 1, 2, \ldots, k$:

$$K(x, X)u_i + \gamma_i = 0, \tag{19}$$

where K is any arbitrary kernel, $u_i \in \mathbb{R}^m$ and $\gamma \in \mathbb{R}$. The primal QPP of F-LS-TWSVC (9) is modified in least squares sense as follows for $i = 1, 2, \ldots, k$:

$$\begin{aligned} \underset{(u_i,\,\gamma_i,\,\eta_i)}{\text{Min}} \ & \frac{1}{2}\|((S_iK(X_i, X)u_i) + \gamma_i e)\|_2^2 + \frac{v}{2}\left(||u_i||_2^2 + \gamma_i^2\right) + C\eta_i^T\eta_i \\ \text{s.t.} \ & |((\overline{S_i}K(\overline{X_i}, X)u_i) + \gamma_i e)| + \eta_i = e. \end{aligned} \tag{20}$$

Similar to the linear case, for each $i = 1, 2, \ldots, k$ the above problem is also decomposed into series of quadratic subproblems where the index of subproblems is $j = 0, 1, 2\ldots$, and solution of which can be derived to be:

$$\left[u_i^{j+1}; \gamma_i^{j+1}\right] = C\left(E_1^{\mathrm{T}}E_1 + Iv + CE_2^{\mathrm{T}}E_2\right)^{-1}E_2^{\mathrm{T}}F^T e, \tag{21}$$

where $E_1 = [S_i(K(X_i, X)) \quad e]$, $E_2 = [\overline{S_i}(K(\overline{X_i}, X)) \quad e]$ and $F = \mathrm{diag}(\mathrm{sign}(\ E_2[u_i^j; b_i^j]))$.

The overall algorithm remains same as of linear case except that now we solve for k kernel-generated surfaces parameters μ_i, γ_i, $i = 1, 2, \ldots, k$.

It can be noted that the nonlinear algorithm requires the solution of (21) which involves calculating the inverse of matrix of order $(m+1)\times(m+1)$. However, we show that (21) can be solved by calculating inverses of two smaller dimension matrices as compare to $(m+1)\times(m+1)$ by using Sherman–Morrison–Woodbury (SMW) [22] formula. Therefore, inversion of matrices in (21) can be further solved by

$$\left[u_i^{j+1}; \gamma_i^{j+1}\right] = C\left(Y - YE_1^{\mathrm{T}}(I + E_1YE_1^{\mathrm{T}})^{-1}E_1Y\right)E_2^{\mathrm{T}}F^T e, \tag{22}$$

where $Y = \frac{1}{v}(I - E_2^{\mathrm{T}}(\frac{v}{C} + E_2E_2^{\mathrm{T}})^{-1}E_2)$, which involves matrix inverses of $(m_i \times m_i)$ and $((m - m_i)\times(m - m_i))$, respectively, for $i = 1, 2, \ldots, k$.

5 Experimental results

The TWSVC, LS-TWSVC and F-LS-TWSVC clustering methods were implemented by using MATLAB 8.1 [23] running on a PC with Intel 3.40 GHz with 16 GB of RAM. The methods were evaluated on several benchmark datasets from UCI Machine Learning Repository [24].

5.1 Performance measure for UCI datasets

To compare the performance of various clustering algorithm, we have used the metric accuracy [17] as the performance criteria for UCI datasets. Given the k *th* cluster labels y_i where $i = 1, \ldots, m$, compute the corresponding similarity matrix $M \in R^{m\times m}$, where

$$M(i,j) = \begin{cases} 1 & : if \quad y_i = y_j \\ 0 & : \text{otherwise.} \end{cases} \tag{23}$$

Let M_t is the similarity matrix computed by the true cluster label of the dataset and M_p corresponds to the label computed from the prediction of clustering method. Then, the metric accuracy of the clustering method is defined as the

$$\text{MetricAccuracy} = \frac{n_{00} + n_{11} - m}{m^2 - m} \times 100\,\%, \tag{24}$$

where n_{00} is the number of zeros in M_p and M_t, and n_{11} is the number of ones in M_p and M_t respectively.

5.2 Performance measure for BSD

To establish the validity of our proposed formulations, we also perform experiments on the Berkeley Segmentation Dataset (BSD) [25] and for comparison we have used $F-$measure [26] and error rate (ER) [27] as the performance criteria.

- $F-$measure can be calculated as

$$F-\text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{25}$$

with respect to human ground truth boundaries. Here,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

- ER can be calculated as

$$\text{ER} = \frac{\text{FP} + \text{FN}}{\text{TT}}, \tag{26}$$

where TP is number of true detection object pixels, FP is the number of false-detection object pixels, FN is the

number of false-detection not object pixels and TT is the total number of pixels present in the image.

For our simulations, we have considered RBF kernel and the values of parameters such as C, v and sigma (kernel parameter) are optimized from the set of values $\{2^i | i = -9, -8, \ldots, 0\}$ using cross-validation methodology [28]. The initial cluster labels and fuzzy membership values are optimized from FNNG initialization as discussed in Sect. 5.3.

5.3 Steps involved in initialization of initial fuzzy membership matrix via fuzzy NNG

Traditionally, the initial labels of clustering can be generated randomly. However, in our algorithm discussed in Algorithm 1, we use fuzzy membership matrix as initial input. In [17], authors have shown via experiments that the results of plane-based clustering methods strongly depend on the initial input of class labels. Hence taking motivation from initialization algorithm based on NNG [17], we implement fuzzy NNG (FNNG) and provide output in the form of fuzzy membership matrix from FNNG method as the initial input to our algorithm. The main process of calculating FNNG is as follows:

1. For the given dataset and a parameter p, construct p nearest neighbor undirected graph whose edges represents the distance between x_i ($i = 1,...,m$) and its p nearest neighbor.
2. From the graph, t clusters are obtained by associating the nearest samples. Further, construct a fuzzy membership matrix S_{ij} where $i = 1, \ldots m$ and $j = 1, \ldots t$ whose (i, j) entry can be calculated as follows,

$$S_{ij} = \frac{1}{d_{ij}}, \tag{27}$$

   where d_{ij} is the euclidean distance of the sample i with the j th cluster. If the current number of cluster t is equal to k, then stop. Else, go to step 3 or 4 accordingly.
3. If $t < k$, disconnect the two connected samples with the largest distance and go to step 2.
4. If $t > k$, compute the Hausdorff distance [29] between every two clusters among the t clusters and sort all pairs in ascending order. Merge the nearest pair of clusters into one, until k clusters are formulated, where the Hausdorff distance between two sets S_1 and S_2 of sample is defined as

$$h(S_1, S_2) = max\{\max_{i \in S_1}\{\min_{j \in S_2}||i-j||\}, \{\max_{i \in S_2}\{\min_{j \in S_1}||i-j||\}\}. \tag{28}$$

For solving F-LS-TWSVC via CCCP, we would need to initialize $[w_1^0 \quad b_1^0]$ and to obtain the same we observe the similarity between F-LS-TWSVC and F-LS-TWSVM, i.e., once the labels are known, solving F-LS-TWSVC is same as solving F-LS-TWSVM [30]. Thus, the initial value of $[w_1^0 \quad b_1^0]$ can be obtained as the solution of F-LS-TWSVM classifier.

5.4 Computational complexity

In [11], authors have shown that TWSVM is approximately 4 times faster than SVM. The computational complexity of TWSVM is $(m^3/4)$, where m is the total number of training samples. In [18], authors have shown that the solution of LS-TWSVM requires system of linear equations to be solved as opposed to the solution of TWSVM which requires system of linear equations along with two QPPs to be solved.

On the similar lines, our algorithm F-LS-TWSVC essentially differs from TWSVC from the optimization problem involved, i.e., in order to obtain k cluster plane parameters, we solve only two matrices inverse of size $(n+1) \times (n+1)$ in linear case, whereas TWSVC seeks to solve system of linear equations along with two QPPs. Table 6 shows the training time comparison among different algorithms with linear kernel on UCI dataset.

For nonlinear F-LS-TWSVC, solution requires inverse of the matrices with order $(m+1)$ which can further be solved by (22) using SMW formula where we tend to solve inverse of two smaller dimension $(m_i \times m_i)$ and $((m - m_i) \times (m - m_i))$ matrices. Table 7 shows the training time comparison among different techniques with nonlinear kernel on UCI dataset. Table 9 shows the training time comparison among different techniques for image segmentation on BSD dataset.

5.5 Experimental results on UCI datasets

In this section, we perform experiments on different UCI datasets with TWSVC, and compared its efficacy with proposed algorithms LS-TWSVC and F-LS-TWSVC, respectively. The summary of UCI datasets is given in Table 1.

In [17], authors have reported clustering accuracy by considering whole dataset for learning the cluster hyperplanes. However, in our presentation of results, we have calculated training clustering accuracy as well as out of sample testing clustering accuracy along with reporting clustering accuracy on the whole dataset together. As a result, we have presented the results in two subsection discussed below.

5.5.1 Part 1 results

Tables 2 and 3 summarize the clustering accuracy results of proposed algorithms F-LS-TWSVC and LS-TWSVC along with TWSVC on several UCI benchmark datasets using linear and nonlinear kernel, respectively. These tables show that metric accuracy of LS-TWSVC and TWSVC are comparable to each other, which further increases approximately 2–5 % on each datasets after incorporating fuzzy membership matrix. In Tables 1 and 2, we have taken results of kPC [8], PPC [9] and FCM [31] from [17] from their respective references.

Figure 1 shows the relations between the parameters and the clustering accuracy (vertical axis) of linear F-LS-TWSVC on the above datasets. It can be found from Fig. 1 that the accuracy of F-LS-TWSVC is affected by both p and c, and higher accuracy is reached by smaller value of p for most datasets.

Figure 2 shows the relations between the parameters and the accuracy (vertical axis) of nonlinear F-LS-TWSVC only on two datasets. In Fig. 2, the x-axis, y-axis and z-axis correspond to the kernel parameter, "c" parameter and clustering accuracy on datasets, respectively. From Fig. 2, it can be found that F-LS-TWSVC performs better for $c < 1$ and $\sigma < 1$. From experiments we have observed that F-LS-TWSVC is invariant from the value of v.

Table 1 Summary of UCI datasets

Dataset	No. of instances	No. of features	No. of classes
Zoo	101	17	7
Wine	178	13	3
Iris	150	4	3
Glass	214	9	6
Dermatology	358	34	6
E. coli	327	7	5
Compound	399	2	2
Haberman	306	3	2
Libras	360	90	15
Page blocks	5473	10	5
Optical recognition	5620	64	10

Table 2 Clustering accuracy with linear kernel on UCI datasets

Data	kPC [8]	PPC [9]	FCM [31]	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	23.31	86.85	85.82	88.83	89.40	92.65
Wine	33.80	73.29	71.05	89.19	89.36	93.18
Iris	50.56	83.68	87.97	91.24	91.02	95.74
Glass	50.65	65.71	71.17	68.08	67.88	69.02
Dermatology	60.50	62.98	69.98	87.89	86.31	91.44
E. coli	27.01	64.42	78.97	83.68	84.04	88.13
Compound	67.54	76.92	84.17	88.31	88.33	88.70
Haberman	60.95	60.95	49.86	62.21	62.14	62.21
Libras	49.90	81.37	51.89	89.42	89.64	90.14
Page blocks	–	–	–	79.88	79.58	81.01
Optical recognition	–	–	–	79.26	79.22	80.17

Table 3 Clustering accuracy with nonlinear kernel on UCI datasets

Data	kPC	PPC	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	89.31	87.84	90.63	91.88	95.14
Wine	55.77	83.05	91.24	91.42	94.66
Iris	77.77	91.24	91.24	91.66	96.66
Glass	63.45	66.95	69.04	69.08	70.96
Dermatology	64.71	71.83	89.44	89.96	93.22
E. coli	86.35	70.17	85.45	87.01	90.17
Compound	88.49	96.84	97.78	96.32	97.88
Haberman	61.57	61.57	61.26	62.14	62.74
Libras	85.32	87.79	90.08	90.56	92.01
Page blocks	–	–	80.78	80.42	82.38
Optical recognition	–	–	81.32	81.06	82.14

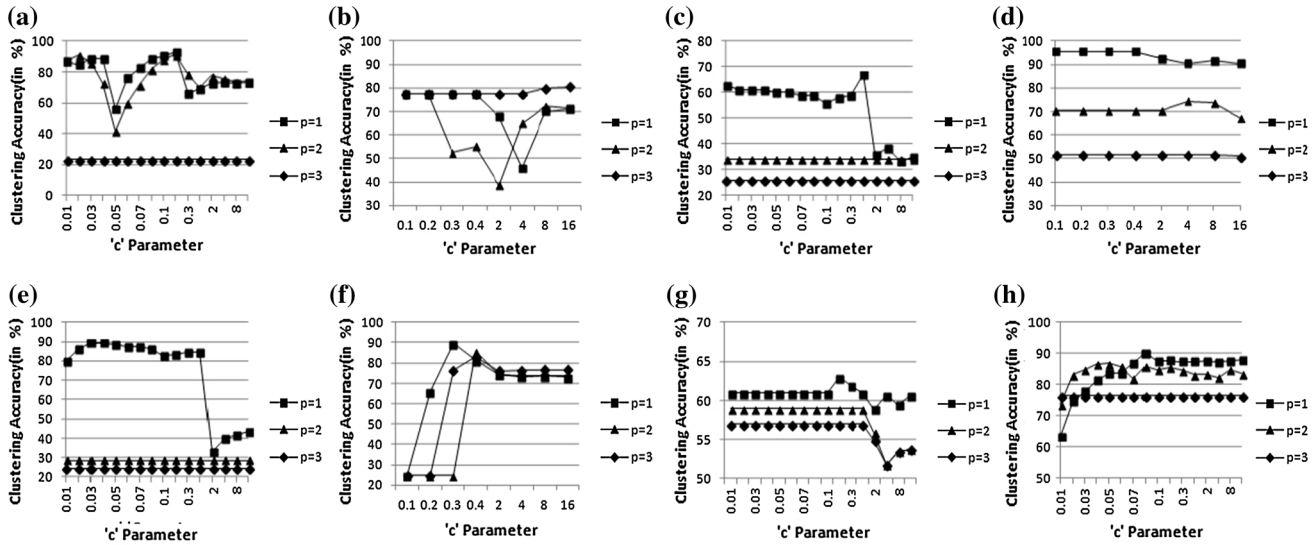

Fig. 1 Illustration of the effectiveness of linear F-LS-TWSVC on UCI datasets with different parameter: **a** zoo, **b** iris, **c** glass, **d** dermatology, **e** *E. coli*, **f** compound, **g** Haberman and **h** libras

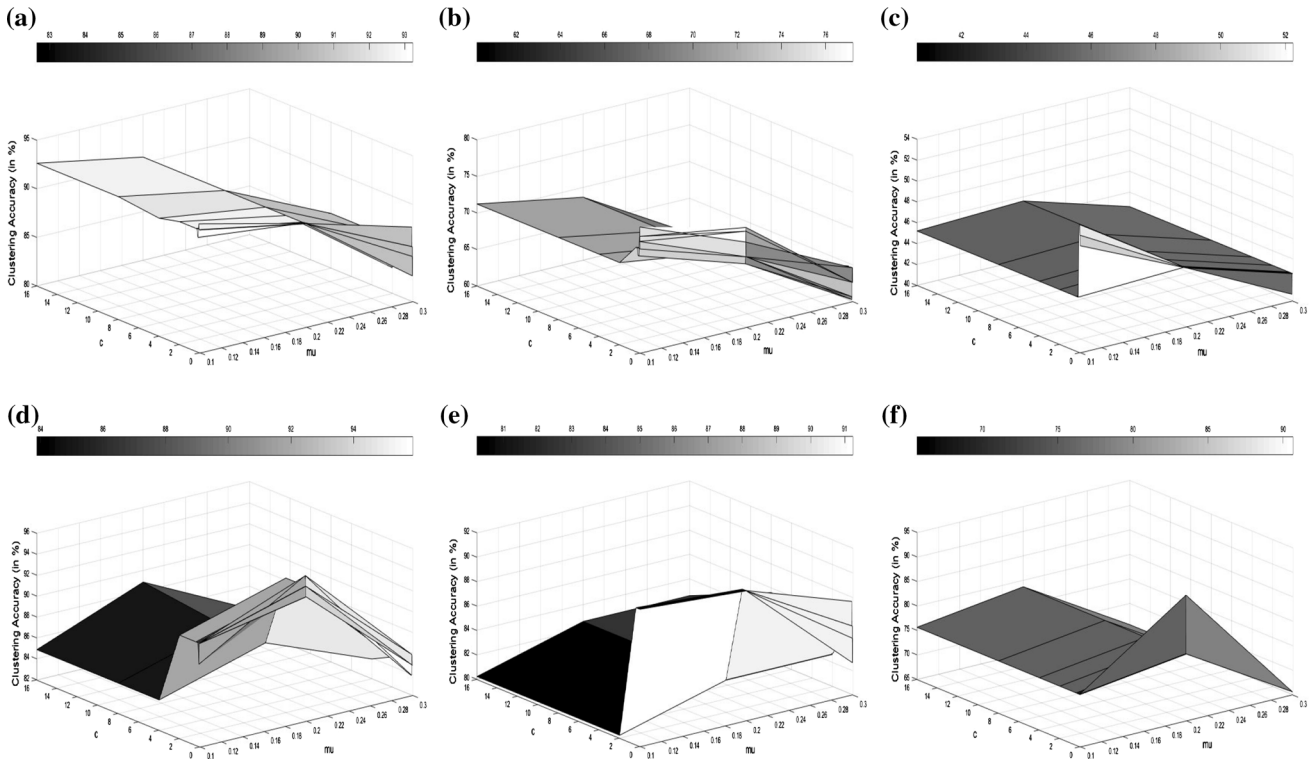

Fig. 2 Illustration of the effectiveness of nonlinear F-LS-TWSVC with different parameter. For Dermatology dataset **a–c** and for Zoo dataset **d–f**

5.5.2 Part 2 results

In this part, clustering accuracy was determined by following the standard fivefold cross-validation methodology [28]. Tables 4 and 5 summarize testing clustering accuracy results of our proposed algorithms F-LS-TWSVC and LS-TWSVC along with TWSVC on several UCI benchmark datasets (Tables 6, 7).

Table 4 Testing clustering accuracy with linear kernel on UCI datasets

Data	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	92.21 ± 3.23	93.56 ± 2.88	96.10 ± 2.18
Wine	85.88 ± 4.16	84.94 ± 4.89	90.92 ± 2.78
Iris	86.01 ± 8.15	86.57 ± 8.05	96.55 ± 1.23
Glass	65.27 ± 4.12	61.20 ± 5.26	65.41 ± 3.80
Dermatology	87.80 ± 2.39	88.08 ± 1.17	92.68 ± 2.42
E. coli	80.96 ± 5.16	82.45 ± 4.96	86.23 ± 4.56
Compound	89.34 ± 3.53	90.70 ± 3.20	90.22 ± 3.29
Haberman	62.57 ± 4.06	60.63 ± 3.94	64.63 ± 3.94
Libras	87.31 ± 1.53	87.34 ± 0.64	88.52 ± 0.49
Page blocks	74.98 ± 4.07	74.63 ± 3.89	76.32 ± 3.12
Optical recognition	74.01 ± 4.78	73.33 ± 5.04	77.40 ± 4.32

Table 5 Testing clustering accuracy comparison with nonlinear kernel on UCI datasets

Data	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	93.47 ± 3.96	94.76 ± 3.04	97.26 ± 2.68
Wine	87.66 ± 4.46	88.04 ± 4.98	92.56 ± 3.48
Iris	88.08 ± 7.45	89.77 ± 7.88	97.25 ± 2.23
Glass	67.27 ± 4.62	64.64 ± 5.66	68.04 ± 4.14
Dermatology	88.26 ± 3.49	88.77 ± 1.74	94.78 ± 2.90
E. coli	83.28 ± 5.46	84.74 ± 5.07	88.96 ± 5.24
Compound	90.14 ± 3.68	90.98 ± 3.44	91.88 ± 3.55
Haberman	62.16 ± 4.26	60.03 ± 3.14	63.36 ± 3.44
Libras	88.16 ± 1.98	88.46 ± 1.06	90.05 ± 0.84
Page blocks	76.68 ± 5.22	75.99 ± 6.07	79.88 ± 5.51
Optical recognition	75.82 ± 5.78	75.32 ± 6.03	78.44 ± 4.11

Table 6 Average training time (in s) with linear kernel on UCI datasets

Data	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	0.1262	0.0042	0.0052
Wine	0.0916	0.0033	0.0047
Iris	0.1645	0.0044	0.0051
Glass	0.2788	0.0062	0.0074
Dermatology	0.2666	0.0114	0.0160
E. coli	0.2687	0.0115	0.0136
Compound	0.5570	0.0199	0.0225
Haberman	0.1156	0.0054	0.0068
Libras	0.4592	0.0319	0.0491
Page blocks	7.6533	0.5316	0.8183
Optical recognition	8.3640	0.1860	0.2220

Table 7 Average training time (in s) with nonlinear kernel on UCI datasets

Data	TWSVC	LS-TWSVC	F-LS-TWSVC
Zoo	1.1200	0.5300	0.7000
Wine	1.6272	0.8447	0.9677
Iris	1.0535	0.5314	0.6468
Glass	6.8200	2.1500	2.6000
Dermatology	12.1500	6.2700	6.9100
E. coli	6.6280	2.9400	3.5111
Compound	17.6589	4.8600	5.3526
Haberman	3.1300	0.9593	1.1900
Libras	28.7700	19.0800	19.9400
Page blocks	204.6000	64.5000	78.6512
Optical recognition	420.5000	190.4100	210.3333

5.6 Experimental results on BSD datasets

In this section, we perform image segmentation on BSD dataset with proposed algorithm F-LS-TWSVC. Texture feature is one of the common feature used in image segmentation. Hence, we extract pixel-level texture feature from the images with the help of Gabor filter. Gabor filter [32] is a class of filters in which a filter of arbitrary orientation and scale is synthesized as linear combination of a set of "basis filter." It allows one to adaptively "steer" a filter to any orientation and scale, and to determine analytically the filter output as a function of orientation and scale. In our experiments, we use three level of scale (0.5, 1.0, 2.0) and four level of orientation $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$. As a result, we have $12(3 \times 4)$ coefficients for each pixel of image. Finally, we use the maximum (in absolute value) of the 12 coefficients for each pixels which represents the pixel-level wise Gabor features of an image. Further, this feature used as an input to FNNG which give us initial membership matrix for every pixels in different clusters. We have also use this Gabor filter to identify number of clusters present in the image.

Table 8 compares the performance of F-LS-TWSVC with TWSVC methods on Berkeley Segmentation Dataset. It is noticeable that for better segmentation, the value of $F-$measure should be high and the value of ER should be lower. Table 8 shows that the value of F-measure for F-LS-TWSVC is higher and the value of ER is lower than TWSVC (Table 9).

Figures 3, 4, 5 and 6 show the segmentation results with F-LS-TWSVC and TWSVC, respectively.

Table 8 F-measure and error rate on BSD dataset

Image-ID	F-measure		ER	
	TWSVC	F-LS-TWSVC	TWSVC	F-LS-TWSVC
3096	0.0250	0.0279	0.0538	0.0499
35070	0.0182	0.0427	0.2216	0.2001
42049	0.0215	0.0699	0.1249	0.0879
71046	0.0619	0.0625	0.2353	0.2280
86016	0.0491	0.0618	0.4806	0.3951
135069	0.0101	0.0141	0.0426	0.0380
198023	0.0500	0.0522	0.0742	0.0687
296059	0.0341	0.0369	0.0645	0.0616

Table 9 Training time (in s) comparison of Image segmentation on BSD dataset

Image-ID	TWSVC	F-LS-TWSVC
3096	060.7049	020.6946
35070	168.8547	130.7123
42049	821.8520	510.9260
71046	118.2998	066.9686
86016	221.4578	130.2693
135069	395.4747	188.3213
198023	482.3645	200.9582
296059	275.1587	185.7195

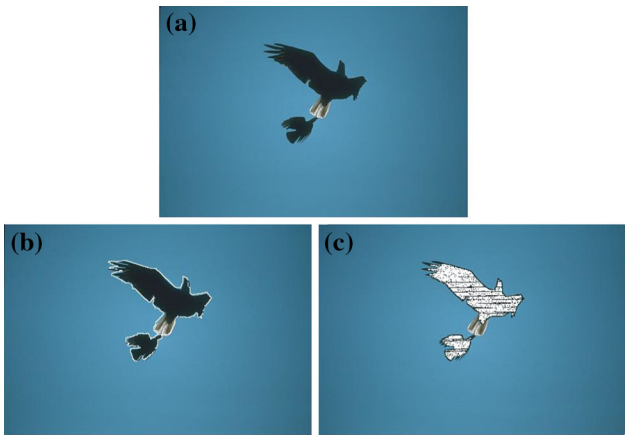

Fig. 3 Segmentation results **a** original image (ImageID-296059), **b** segmented image with F-LS-TWSVC and **c** segmented image with TWSVC

6 Conclusions

In this paper, we have extended TWSVM-based clustering method TWSVC to least squares version termed as LS-TWSVC for both linear and nonlinear cases. Further extension of LS-TWSVC to fuzzy scenario by introducing

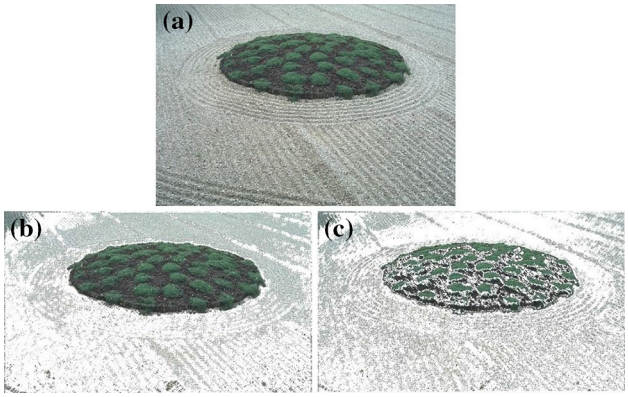

Fig. 4 Segmentation results **a** original image (ImageID-86016), **b** segmented image with F-LS-TWSVC and **c** segmented image with TWSVC

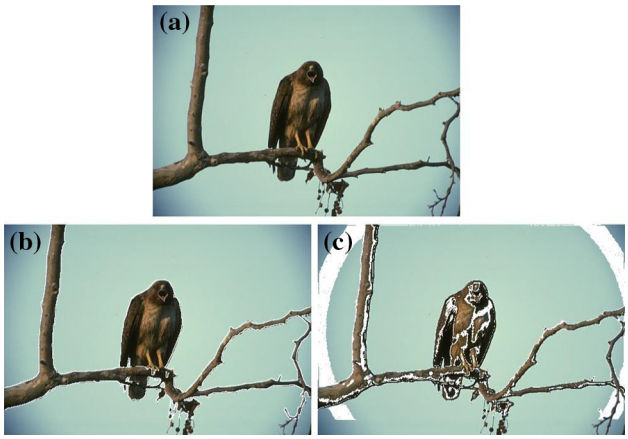

Fig. 5 Segmentation results **a** original image (ImageID-71046), **b** segmented image with F-LS-TWSVC and **c** segmented image with TWSVC

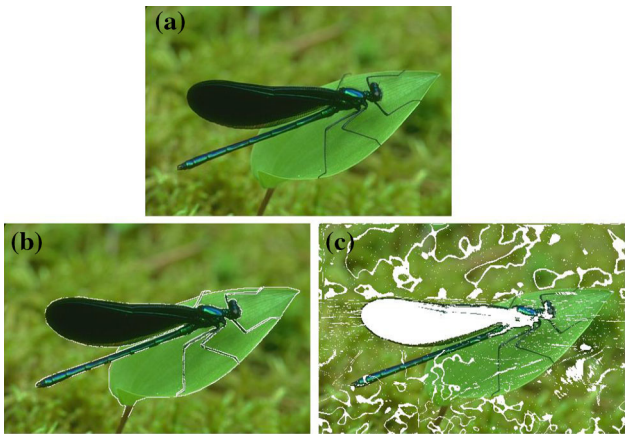

Fig. 6 Segmentation results **a** original image (ImageID-198023), **b** segmented image with F-LS-TWSVC and **c** segmentation results with TWSVC

a fuzzy membership matrix has been considered. The proposed algorithms yields k cluster center planes by solving a series of system of linear equations as opposed to solving series of QPPs along with system of linear equations required for TWSVC algorithm. The experimental results on several UCI benchmark datasets shows that our proposed method achieves better clustering accuracy to that of TWSVC and with considerably less computational time. We have also validated our algorithm for image segmentation where the images have been considered from BSD image dataset.

Currently in FNNG initialization method, we have used $1/d_{ij}$ membership function where d_{ij} represent the distance between two clusters i and j which would not be suitable for noisy data, i.e., Haberman dataset. Therefore, in the future work, we would like to contribute by adding other robust membership function which could handle the noisy data as well.

Acknowledgments We are very thankful to Mr. Keshav Goyal for his initial contribution to the analysis of the draft. We are also extremely grateful to the anonymous reviewers and Editor for their valuable comments that helped us to enormously improve the quality of the paper.

References

1. Anderberg M (1973) Cluster analysis for applications. Academic Press, New York
2. Jain A, Murty M, Flynn P (1999) Data clustering: a review. ACM Comput Surv CSUR 31(3):264–323
3. Qimin C, Qiao G, Yongliang W, Xianghua W (2015) Text clustering using VSM with feature clusters. Neural Comput Appl 26(4):995–1003
4. Zhan Y, Yin J, Liu X (2013) Nonlinear discriminant clustering based on spectral regularization. Neural Comput Appl 22(7–8):1599–1608
5. Tu E, Cao L, Yang J, Kasabov N (2014) A novel graph-based k-means for nonlinear manifold clustering and representative selection. Neurocomputing 143:109–122
6. Liu X, Li M (2014) Integrated constraint based clustering algorithm for high dimensional data. Neurocomputing 142:478–485
7. Bradley P, Mangasarian O (1997) Clustering via concave minimization. Adv Neural Inf Process Syst 9:368–374
8. Bradley P, Mangasarian O (2000) k-Plane clustering. J Glob Optim 16(1):23–32
9. Shao Y, Bai L, Wang Z, Hua X, Deng N (2013) Proximal plane clustering via eigenvalues. Procedia Comput Sci 17:41–47
10. Yang Z, Guo Y, Li C, Shao Y (2014) Local k-proximal plane clustering. Neural Comput Appl 26(1):199–211
11. Jayadeva KR, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29:905–910
12. Mehrkanoon S, Huang X, Suykens JA (2014) Non-parallel support vector classifiers with different loss functions. Neurocomputing 143:294–301
13. Xie X, Sun S (2014) Multi-view Laplacian twin support vector machines. Appl Intell 41:1059–1068
14. Xie X, Sun S (2015) Multitask centroid twin support vector machines. Neurocomputing 149:1085–1091
15. Ding S, Zhang N, Zhang X, Wu F (2016) Twin support vector machine: theory, algorithm and applications. Neural Comput Appl. doi:10.1007/s00521-016-2245-4
16. Tian Y, Qi Z (2014) Review on: twin support vector machines. Ann Data Sci 1:253–277
17. Wang Z, Shao Y, Bai L, Deng N (2014) Twin support vector machine for clustering. IEEE Tran Neural Netw Learn Syst. doi:10.1109/TNNLS.2014.2379930
18. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings of seventh international conference on knowledge and data discovery, pp 77–86
19. Keller JM, Gray MR, Givens JA (1985) A fuzzy k-nearest neighbor algorithm. IEEE Trans Syst Man Cybern 4:580–585
20. Yuille AL, Rangarajan A (2002) The concave-convex procedure (CCCP). Adv Neural Inf Process Syst 2:1033–1040
21. Kumar A, Gopal M (2009) Least squares twin support vector machines for pattern classification. Exp Syst Appl 36:7535–7543
22. Golub GH, Van Loan CF (1996) Matrix Computations. The John Hopkins University Press, Baltimore
23. MATLAB (1994–2001) User's guide. The MathsWorks, Inc. http://www.mathworks.com
24. Blake CL, Merz CJ (1998) UCI Repository for machine learning databases. University of California, Department of Information and Computer Sciences, Irvine. http://www.ics.uci.edu/~mlearn/MLRepository.html
25. Arbelaez P, Fowlkes C, Martin D (2007) The berkeley segmentation dataset and benchmark. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds
26. Mehrkanoon S, Alzate C, Mall R, Langone R, Suykens J (2015) Multiclass semisupervised learning based upon kernel spectral clustering. IEEE Trans Neural Netw Learn Syst 26(4):720–733
27. Wang XY, Wang T, Bu J (2011) Color image segmentation using pixel wise support vector machine classification. Pattern Recognit 44(4):777–787
28. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
29. Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal Mach Intell 15(9):850–863
30. Sartakhti JS, Ghadiri N, Afrabandpey H (2015) Fuzzy Least squares twin support vector machines. arXiv:1505.05451
31. Wang X, Wang Y, Wang L (2004) Improving fuzzy c-means clustering based on feature-weight learning. Pattern Recognit Lett 25:1123–1132
32. Manjunath BS, Ma WY (1996) Texture features for browsing and retrieval of image data. IEEE Trans Pattern Anal Mach Intell 18(8):837–842