Neural Computing and Applications (2020) 32:9885–9896
https://doi.org/10.1007/s00521-019-04511-3

ORIGINAL ARTICLE

Ramp-based twin support vector clustering

Zhen Wang[1] · Xu Chen[1] · Yuan-Hai Shao[2] · Chun-Na Li[2]

Received: 21 March 2019 / Accepted: 21 September 2019 / Published online: 8 October 2019

Abstract

Traditional plane-based clustering methods measure the within-cluster or between-cluster scatter by linear, quadratic or some other unbounded functions, which are sensitive to the samples far from the cluster center. This paper introduces the ramp functions into plane-based clustering and proposes a ramp-based twin support vector clustering (RampTWSVC). RampTWSVC is very robust to the samples far from the cluster center, because its within-cluster and between-cluster scatters are measured by the bounded ramp functions. Thus, it is easier to find the intrinsic clusters than other plane-based clustering methods. The nonconvex programming problem in RampTWSVC is solved efficiently through an alternating iteration algorithm, and its local solution can be obtained in a finite number of iterations theoretically. In addition, its nonlinear manifold clustering formation is also proposed via a kernel trick. Experimental results on several benchmark datasets show the better performance of our RampTWSVC compared with other plane-based clustering methods.

Keywords Clustering · Plane-based clustering · Ramp function · Twin support vector clustering · Nonconvex programming

1 Introduction

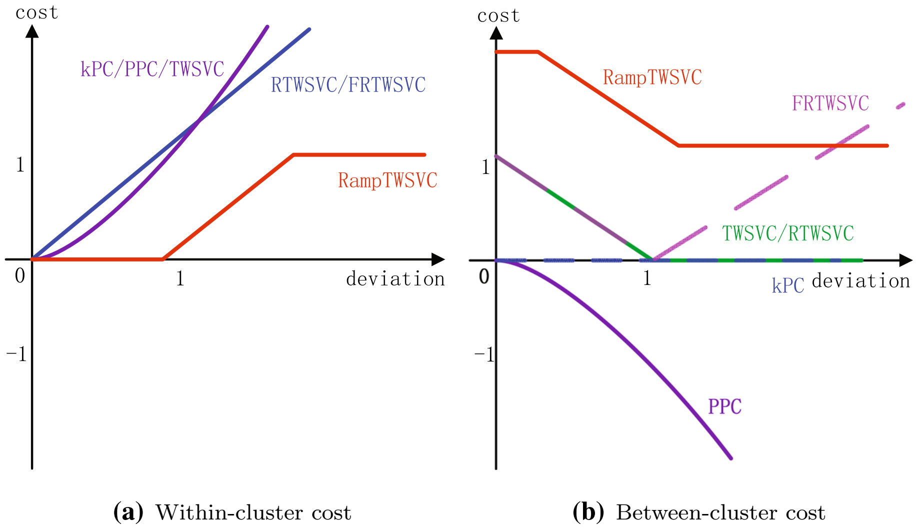
Clustering that discovers the relationship among data samples has been applied to many real-world problems, e.g., marketing, text mining and web analysis [1–7]. In particular, the partition-based clustering methods [1, 8, 9] are widely used in real applications for their simplicity, e.g., the classical kmeans [10]. The partition-based clustering supposes there are k clusters and each cluster has a cluster center. Then, the data samples are assigned to these clusters according to the distances from the cluster centers. Kmeans supposes the cluster centers are some points in the sample space, but the cluster centers may be some other types in the datasets, e.g., planes in the "cross-planes" dataset [11–13]. As an extension of point center, the plane center is able to discover comprehensive structures in the sample space. Therefore, the partition-based clustering with planes as cluster centers, called plane-based clustering, has been studied widely [14–20].

In plane-based clustering, starting from an initialization, the clustering result is obtained iteratively through the cluster assignment and cluster update steps. In the cluster assignment step, the samples are assigned to the clusters by the distances from the cluster center planes. In the cluster update step, the cluster center planes are the solutions of k optimization problems, where different plane-based methods have different optimization problems. In general, the optimization problems consist of two aspects: within-cluster compactness and between-cluster separation. The first proposed k-plane clustering (kPC) [14] measured the within-cluster scatter by a quadratic function, and another proximal planes clustering (PPC) [15, 16] measured the within-cluster and between-cluster scatters by the same quadratic function. Subsequently, twin support vector clustering (TWSVC) [17] improved the between-cluster scatter by hinge function [12, 21], robust twin support vector clustering (RTWSVC) [18] and fast robust twin support vector k-plane clustering (FRTWSVC) [18] improved them by absolute function. Figure 1 exhibits the metric functions used in these methods. Since these plane-based clustering methods measure the within-cluster and/or between-cluster scatter by some unbounded functions, some samples may influence the cluster center planes

✉ Yuan-Hai Shao
shaoyuanhai21@163.com

[1] School of Mathematical Sciences, Inner Mongolia University, Hohhot 010021, People's Republic of China

[2] School of Management, Hainan University, Haikou 570228, People's Republic of China



(a) Within-cluster cost (b) Between-cluster cost

Fig. 1 Metric functions used in kPC, PPC, TWSVC, RTWSVC, FRTWSVC and RampTWSVC to measure the within-cluster and between-cluster scatters. The horizontal axis denotes the deviation of a sample from the cluster center, and the vertical axis denotes the cost to fit that sample. When a cost value is negative, the cost becomes a reward

significantly, e.g., the samples far from their corresponding cluster center.

To further reduce the influence of the samples far from the cluster center, a better choice is to hire some bounded functions to measure the within-cluster and between-cluster scatters instead of unbounded functions. Therefore, we consider to hire the ramp functions [22] into the plane-based clustering, where the ramp functions are bounded piecewise linear functions and have been applied in semi-supervised and supervised learning successfully [23–25]. In this paper, a ramp-based twin support vector clustering method, called RampTWSVC, is proposed by introducing the ramp functions for the measurement of the within-cluster and between-cluster scatters (see Fig. 1). RampTWSVC reduces the influence of the samples far from the cluster centers to obtain the stable cluster center planes. The optimization problems in our RampTWSVC are some more complicated nonconvex programming problems than other plane-based clustering methods. These problems are recast to some mixed-integer programming problems with the same formation. Then, an iterative algorithm is proposed to solve the mixed-integer programming problem, and we prove that the iterative algorithm terminates in a finite number of iterations at a local solution. In addition, RampTWSVC is extended to nonlinear case via a kernel trick to cope with the manifold clustering [26, 27]. Experimental results on the benchmark datasets show the better performance of the proposed RampTWSVC compared with other plane-based clustering methods.

2 Review of plane-based clustering

In this paper, we consider m data samples $\{x_1, x_2, \ldots, x_m\}$ in the n-dimensional real vector space $\mathbb{R}^n$. Assume these m samples belong to k clusters with their corresponding labels $y \in \{1, 2, \ldots, k\}$, and they are represented by a matrix $X = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^{n\times m}$. We further organize the samples from X with the current label i into a matrix X_i and those with the rest labels into a matrix $\hat{X}_i$ with $i = 1, 2, \ldots, k$. For readers' convenience, the symbols X, X_i, and $\hat{X}_i$ will also refer to the corresponding sets, depending on the specific context they appear. For example, the symbol X can be comprehended as a matrix belonging to $R^{n\times m}$ or a set that contains m samples. The cluster center planes are defined as

$$f_i(x) = w_i^\top x + b_i = 0, \ i = 1, \ldots, k, \tag{1}$$

where $w_i \in \mathbb{R}^n$ is the weight vector and $b_i \in \mathbb{R}$ is the bias.

The following plane-based clustering methods share the same clustering procedures. Starting from an initial assignment of the m samples into k clusters, the cluster center planes (1) are constructed by the current cluster assignment. Once the cluster center planes are obtained, the m samples are reassigned by

$$y = \arg\min_i |w_i^\top x + b_i|, \tag{2}$$

where $|\cdot|$ denotes the absolute value. The cluster center planes and the sample labels are updated alternately until some termination conditions are satisfied. In the following, we briefly describe the different constructions of the cluster center planes in kPC, PPC, TWSVC and RTWSVC.

2.1 kPC and PPC

kPC [14] wishes each cluster center plane close to the current cluster samples. Further on, PPC [15] considers that it should also be far away from different cluster samples. Therefore, the ith ($i = 1, \ldots, k$) cluster center plane in PPC is constructed by considering the problem

$$\begin{aligned} \min_{w_i,b_i} \; & ||X_i^\top w_i + b_i e||^2 - c||\hat{X}_i^\top w_i + b_i e||^2 \\ \text{s.t.} \; & ||w_i|| = 1, \end{aligned} \tag{3}$$

where $|| \cdot ||$ denotes the L_2-norm, e is a column vector of ones with an appropriate dimension and $c > 0$ is a user-set parameter, while the optimization problem in kPC does not have the second term in the objective of (3).

From the objective of (3), it is obvious that the samples from the within-cluster receive quadratic costs, and the samples from the between-cluster receive quadratic rewards. Therefore, the samples far from the cluster center will have great impact on the potential cluster center planes, and PPC may obtain a cluster center plane which is far from the current cluster.

2.2 TWSVC and RTWSVC

In contrast, TWSVC [17] degrades the rewards of the samples from different clusters by considering the problem with $i = 1, \ldots, k$ as

$$\min_{w_i,b_i} \; \frac{1}{2}||X_i^\top w_i + b_i e||^2 + ce^\top (e - |\hat{X}_i^\top w_i + b_i e|)_+, \tag{4}$$

where $(\cdot)_+$ replaces the negative parts by zeros.

From the second part of (4), it is clear that the samples from the between-cluster would have a certain impact on the cluster center plane only with deviation in [0, 1). Thus, TWSVC is more robust than PPC [17]. However, the issue of within-cluster also exists because of the quadratic function in the first part of (4). Thus, RTWSVC [18] was proposed to decrease its influence by replacing the L_2-norm in (4) with the L_1-norm. Note that the metric function of within-cluster of RTWSVC is unbounded from Fig. 1. In order to eradicate the influence of the samples far from the cluster center, it is reasonable to hire a bounded metric function for the within-cluster, whose principle is similar to the metric function for the between-cluster used in TWSVC.

3 RampTWSVC

Similar to the above plane-based clustering methods mentioned in Sect. 2, our RampTWSVC starts with initial sample labels and then computes each cluster center plane and the sample labels alternately, until some termination conditions are satisfied. In the following, we consider to obtain one of the cluster center planes for the given samples with their labels.

3.1 Formation

To obtain the ith ($i = 1, \ldots, k$) cluster center plane, our RampTWSVC considers the following problem

$$\min_{w_i,b_i} \; c_1 \sum_{x_j \in X_i} R_1(x_j) + c_2 \sum_{x_j \in \hat{X}_i} R_2(x_j) + \frac{1}{2}(||w_i||^2 + ||b_i||^2), \tag{5}$$

where $c_1, c_2 > 0$ are the trade-offs. $R_1(x)$ and $R_2(x)$ are two piecewise linear functions w.r.t. the deviation $|f_i(x)| = |w_i^\top x + b_i|$ (see Fig. 1) as

$$R_1(x) = \begin{cases} 0 & \text{if } |f_i(x)| \leq 1 - \Delta, \\ 1 - s & \text{if } |f_i(x)| \geq 2 - \Delta - s, \\ |f_i(x)| - 1 + \Delta & \text{otherwise}, \end{cases} \tag{6}$$

and

$$R_2(x) = \begin{cases} 2 + 2\Delta & \text{if } |f_i(x)| \leq -s, \\ 1 + \Delta - s & \text{if } |f_i(x)| \geq 1 + \Delta, \\ -|f_i(x)| + 2 + 2\Delta - s & \text{otherwise}, \end{cases} \tag{7}$$

where $\Delta \in [0, 1)$ and $s \in (-1, 0]$ are two constants to control the function form (for instance, $\Delta = 0.3$ and $s = -0.2$ [23]). $||w_i||$ and $||b_i||$ are the regularization terms to control the problem complexity.

It is obvious that both of the functions $R_1(x)$ (for the within-cluster X_i) and $R_2(x)$ (for the between-cluster $\hat{X}_i$) have upper and lower bounds. Thus, the samples much further from the cluster center plane do not have greater impact on the cluster center plane. The above property indicates our RampTWSVC is more robust than PPC, TWSVC and RTWSVC.

In the following, we extend our RampTWSVC to nonlinear manifold clustering, and the solutions to the linear and nonlinear RampTWSVC are elaborated in the next subsection.

The plane-based clustering methods can be extended to nonlinear manifold clustering easily via kernel tricks [13, 28]. For instance, by introducing a pre-defined kernel function $K(\cdot, X)$ [11, 17, 21], the plane-based nonlinear clustering seeks k cluster center manifolds in the kernel-generated space as

$$g_i(x) = K(x, X)^\top w_i + b_i = 0,\ i = 1, \ldots, k. \tag{8}$$

Then, the nonlinear RampTWSVC considers introducing the ramp functions into the plane-based nonlinear clustering. By replacing $f_i(x)$ with $g_i(x)$ in (6) and (7), and substituting them into (5), one can easily obtain the optimization problems to construct the cluster center manifolds (8). When we obtain the k cluster centers (8), a sample x is assigned to a cluster given by

$$y = \arg\min_i |K(x, X)^\top w_i + b_i|. \tag{9}$$

The procedure of the nonlinear case is the same as the linear one, so the details are omitted.

3.2 Solution

In this subsection, we study the solution to problem (5). The corresponding problem in nonlinear RampTWSVC is similar to the one in linear case. For convenience, let $u_i = (w_i^\top, b_i)^\top$, Z_i be a matrix whose jth column z_j is x_j with an additional feature 1 (where the corresponding x_j belongs to the ith cluster), and let $\hat{Z}_i$ be a matrix whose column is similar to z_j (where the corresponding x_j does not belongs to the ith cluster). After some algebra, problem (5) is recast to

$$\begin{aligned}\min_{u_i}\ & \frac{1}{2}||u_j||^2 + c_1 e^\top(-1 + \Delta - Z_i^\top u_i)_+ + c_1 e^\top(-1 + \Delta \\ & + Z_i^\top u_i)_+ + c_2 e^\top(1 + \Delta - \hat{Z}_i^\top u_i)_+ + c_2 e^\top(1 + \Delta \\ & + \hat{Z}_i^\top u_i)_+ - c_1 e^\top(s - 2 + \Delta - Z_i^\top u_i)_+ - c_1 e^\top(s - 2 + \Delta \\ & + Z_i^\top u_i)_+ - c_2 e^\top(s - \hat{Z}_i^\top u_i)_+ - c_2 e^\top(s + \hat{Z}_i^\top u_i)_+.\end{aligned} \tag{10}$$

It is easy to see that the above problem is a nonconvex programming problem because of the concave part $-(\cdot)_+$. By introducing two auxiliary vectors $p_1 \in \{-1, 0, 1\}^{m_i}$ and $p_2 \in \{-1, 0, 1\}^{m - m_i}$ (where m_i is the sample number of the current ith cluster), the above problem is equivalent to the following mixed-integer programming problem

$$\begin{aligned}\min_{u_i, p_1, p_2}\ & \frac{1}{2}||u_i||^2 + c_1 e^\top(-1 + \Delta - Z_i^\top u_i)_+ \\ & + c_1 e^\top(-1 + \Delta + Z_i^\top u_i)_+ + c_2 e^\top(1 + \Delta - \hat{Z}_i^\top u_i)_+ \\ & + c_2 e^\top(1 + \Delta + \hat{Z}_i^\top u_i)_+ + c_1 p_1^\top Z_i^\top u_i + c_2 p_2^\top \hat{Z}_i^\top u_i \\ s.t.\ p_1(j) = & \begin{cases} -1 & \text{if } z_j^\top u_i > 2 - \Delta - s, \\ 1 & \text{if } z_j^\top u_i < -2 + \Delta + s, \forall z_j \in Z_i \\ 0 & \text{otherwise,} \end{cases} \\ p_2(j) = & \begin{cases} -1 & \text{if } z_j^\top u_i > -s, \\ 1 & \text{if } z_j^\top u_i < s, \quad \forall z_j \in \hat{Z}_i \\ 0 & \text{otherwise,} \end{cases}\end{aligned} \tag{11}$$

where $p_1(j)$ and $p_2(j)$ are the corresponding jth elements of p_1 and p_2, respectively.

Here, we propose an alternating iteration algorithm to solve mixed-integer programming problem (11). Starting with an initialized $u_i^{(0)}$, it is easy to calculate $p_1^{(0)}$ and $p_2^{(0)}$ by the constraints of (11). For fixed $p_1^{(t-1)}$ and $p_2^{(t-1)}$ $(t = 1, 2, \ldots)$, (11) becomes an unconstrained convex problem and its solution can be obtained by many algorithms easily (e.g., sequential minimal optimization (SMO) [29] and fast Newton-Amijio algorithm [30]). Once $u_i^{(t)}$ is obtained, $p_1^{(t)}$ and $p_2^{(t)}$ are updated again. The loop will be continued until the objective of (11) does not decrease any more.

Theorem 3.1 *The above alternating iteration algorithm terminates in a finite number of iterations at a local optimal point, where a local optimal point of the mixed-integer programming problem (11) is defined as the point (u_i^*, p_1^*, p_2^*) if u_i^* is the global solution to problem (11) with fixed (p_1^*, p_2^*) and vice versa.*

Proof From the procedure of the alternating iteration algorithm, it is obvious that the global solutions to (11) with fixed u_i or (p_1, p_2) are obtained in iteration. Since

Table 1 Details of the benchmark datasets, where m is the number of samples, n is the number of dimensions and k is the number of classes

	Data	m	n	k
(a)	Arrhythmia	452	278	13
(b)	Dermatology	366	34	6
(c)	*E. coli*	336	7	8
(d)	Glass	214	9	6
(e)	Iris	150	4	3
(f)	Libras	360	90	15
(g)	Seeds	210	7	3
(h)	Wine	178	13	3
(i)	Zoo	101	16	7
(j)	Bupa	345	6	2
(k)	Echocardiogram	131	10	2
(l)	Heartstatlog	270	13	2
(m)	Housevotes	435	16	2
(n)	Ionosphere	351	33	2
(o)	Sonar	208	60	2
(p)	Soybean	47	35	2
(q)	Spect	267	44	2
(r)	Wpbc	198	34	2
(s)	Biodeg	1055	41	2
(t)	Cnae	1080	856	9
(u)	Dna	2000	180	3
(v)	Tic	5822	85	2

Table 2 Linear clustering on benchmark datasets

Data	kmeans	kPC	PPC	TWSVC	RTWSVC	FRTWSVC	Ours
	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)
	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)
(a)	65.72 ± 0.53	32.31	65.20	32.31	32.31	32.31	**79.42**
	19.55 ± 0.99	5.49	6.70	5.49	5.49	5.49	10.10
(b)	69.76 ± 0.77	60.50	70.36	71.93	60.50	60.50	**72.67**
	11.47 ± 2.15	**29.65**	3.48	27.40	28.95	28.95	24.42
(c)	82.19 ± 2.68	33.11	66.46	**85.74**	34.33	34.33	79.42
	56.84 ± 4.42	8.61	9.65	33.43	10.42	10.42	43.35
(d)	65.58 ± 3.22	57.73	**66.75**	66.62	57.59	57.40	62.77
	35.76 ± 2.23	22.55	8.54	35.40	17.69	18.20	20.95
(e)	84.57 ± 6.86	67.54	60.95	91.24	92.67	**94.95**	86.79
	70.47 ± 9.10	25.41	12.04	85.59	82.31	**86.97**	71.71
(f)	**90.84** ± 0.41	89.42	87.93	89.97	89.42	89.42	87.11
	57.50 ± 2.28	56.40	15.84	56.40	56.40	56.40	44.47
(g)	**87.35** ± 0.15	71.80	62.39	63.40	72.24	76.16	74.07
	69.77 ± 0.68	42.43	18.33	51.27	43.17	52.09	45.74
(h)	71.06 ± 1.29	52.73	57.49	66.90	**72.20**	70.26	69.45
	41.97 ± 1.44	7.33	4.70	35.48	**45.35**	41.08	35.16
(i)	87.49 ± 1.96	54.12	84.06	88.83	54.12	54.12	**90.22**
	71.93 ± 3.15	34.23	55.56	73.33	32.15	32.15	**76.98**
(j)	50.39 ± 0.03	50.31	51.13	51.22	53.34	52.10	**55.82**
	0.09 ± 0.02	0.22	0.23	0.42	3.73	1.86	**7.07**
(k)	66.41 ± 7.92	52.81	56.66	56.10	**75.01**	**75.01**	71.84
	24.79 ± 17.27	0.54	2.99	36.87	**39.64**	**39.64**	35.46
(l)	51.45 ± 0.07	50.04	50.35	50.81	51.40	51.40	**51.82**
	1.87 ± 0.07	0.02	0.15	**13.11**	1.63	1.67	2.40
(m)	78.83 ± 0.15	63.77	68.77	75.83	71.40	71.40	**79.61**
	48.07 ± 0.38	34.16	27.27	45.19	39.36	39.36	**50.15**
(n)	58.89 ± 0.00	61.76	53.23	53.85	**67.64**	66.63	61.76
	13.12 ± 0.00	13.00	3.26	21.13	**23.04**	21.26	12.91
(o)	50.22 ± 0.18	49.80	49.99	50.43	51.26	50.06	**51.62**
	0.74 ± 0.28	0.01	0.23	0.01	2.06	0.67	**4.05**
(p)	93.41 ± 13.90	91.67	**100.0**	50.05	91.67	91.67	**100.0**
	86.95 ± 27.53	78.05	**100.0**	1.70	78.05	78.05	**100.0**
(q)	52.97 ± 0.00	65.86	50.67	65.86	50.88	50.58	**67.17**
	8.48 ± 0.00	0.51	0.51	0.51	0.35	0.34	1.15
(r)	56.03 ± 0.00	52.95	57.95	56.03	53.48	57.15	**64.15**
	0.08 ± 0.00	0.21	0.27	0.05	0.01	**2.95**	1.33
(s)	51.52 ± 0.00	50.28	54.14	54.82	50.28	50.28	**55.68**
	0.02 ± 0.00	**5.24**	0.02	0.43	**5.24**	**5.24**	1.66
(t)	**76.20** ± 4.19	11.61	75.38	66.85	11.61	11.61	75.82
	21.67 ± 8.05	1.00	2.17	7.16	1.00	1.00	20.12
(u)	59.46 ± 23.10	64.75	70.53	71.18	64.75	64.75	**76.71**
	31.52 ± 28.71	30.23	34.99	37.83	30.23	30.23	**47.71**
(v)	50.72 ± 0.03	49.89	51.24	49.88	49.88	49.88	**54.54**
	1.27 ± 0.01	0.08	2.03	0.50	2.29	2.29	**6.83**

Table 3 Nonlinear clustering on benchmark datasets

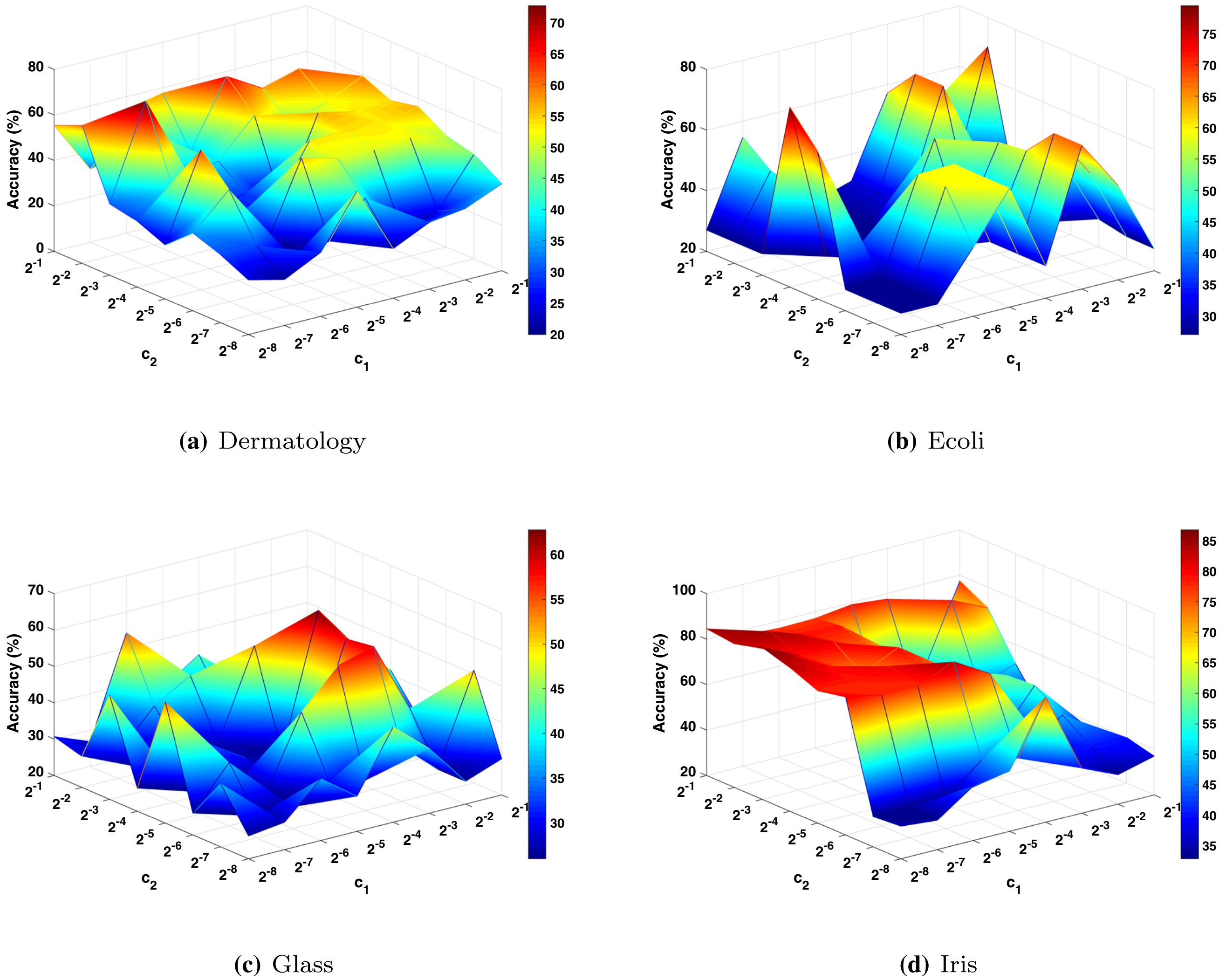
Data	kmeans	kPC	PPC	TWSVC	RTWSVC	FRTWSVC	Ours
	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)	AC (%)
	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)	MI (%)
(a)	47.32 ± 3.08	62.17	**64.82**	46.89	62.17	62.17	62.19
	10.76 ± 1.24	10.14	6.51	9.65	10.14	10.14	8.93
(b)	71.66 ± 1.26	72.60	70.62	72.60	72.60	72.60	**72.90**
	17.84 ± 3.67	18.00	3.65	18.00	18.00	18.00	**26.79**
(c)	79.93 ± 1.24	82.49	69.13	**88.29**	82.49	82.68	83.01
	49.31 ± 2.28	57.79	16.46	**62.21**	57.79	57.57	49.97
(d)	69.27 ± 1.45	69.04	66.82	70.10	69.04	69.04	**70.77**
	37.50 ± 2.09	**41.42**	7.35	23.42	**41.42**	**41.42**	0.2918
(e)	87.63 ± 8.09	91.24	59.47	91.24	91.24	91.24	**94.95**
	76.26 ± 9.85	79.15	13.93	79.15	79.15	79.15	**86.23**
(f)	**90.60** ± 0.42	85.67	88.04	90.08	86.38	86.38	89.60
	54.86 ± 1.24	17.95	17.79	**56.98**	22.28	22.28	51.18
(g)	87.02 ± 0.77	78.41	68.48	81.54	79.03	78.41	**87.14**
	69.74 ± 0.55	58.81	26.95	63.48	54.07	58.81	**69.98**
(h)	52.07 ± 4.07	60.75	**72.55**	44.89	60.75	60.75	64.06
	13.84 ± 3.04	20.35	**41.23**	6.12	20.35	20.35	25.98
(i)	87.14 ± 3.39	90.63	89.52	90.63	90.63	90.63	**91.25**
	70.79 ± 5.39	77.99	72.90	77.99	77.99	77.99	**79.70**
(j)	51.08 ± 0.35	51.22	**53.04**	51.98	51.22	51.22	**53.04**
	0.46 ± 0.42	0.37	2.90	1.60	0.37	0.37	**4.54**
(k)	71.14 ± 0.82	55.04	56.66	56.66	55.04	55.04	**71.84**
	32.41 ± 0.53	0.85	2.73	2.73	0.85	0.85	28.53
(l)	50.83 ± 0.41	53.00	51.54	50.92	53.00	53.00	**54.91**
	1.88 ± 0.54	3.79	1.64	0.81	3.79	3.79	**6.98**
(m)	79.79 ± 0.94	75.50	75.83	**91.21**	75.50	75.50	80.68
	46.91 ± 1.87	42.09	46.38	**72.31**	42.09	42.09	48.86
(n)	62.32 ± 0.00	59.14	59.89	60.67	59.14	59.14	**82.92**
	22.24 ± 0.00	23.79	10.87	13.60	23.79	23.79	**52.32**
(o)	50.16 ± 0.28	51.62	52.66	52.22	51.62	51.62	**54.52**
	0.39 ± 0.39	4.24	4.08	5.43	4.24	4.24	**6.64**
(p)	100.0 ± 0.00	100.0	100.0	100.0	100.0	100.0	100.0
	100.0 ± 0.00	100.0	100.0	100.0	100.0	100.0	100.0
(q)	60.68 ± 4.79	66.73	68.06	68.06	66.73	66.73	**68.98**
	3.38 ± 3.72	0.17	2.35	2.35	0.17	0.17	**17.69**
(r)	63.40 ± 0.52	63.08	**64.15**	63.08	63.08	63.08	63.61
	0.58 ± 0.52	0.25	**1.42**	0.25	0.25	0.25	0.25
(s)	58.54 ± 1.01	56.21	62.35	61.69	63.41	59.68	**66.17**
	5.42 ± 1.91	5.31	15.95	11.13	13.81	10.38	**18.60**
(t)	79.31 ± 1.84	78.75	79.98	74.14	**80.64**	78.75	**80.64**
	12.76 ± 2.03	**21.11**	2.98	11.55	9.32	**21.11**	9.32
(u)	64.65 ± 4.15	66.28	65.48	62.53	66.28	66.28	**67.03**
	20.23 ± 3.43	**30.39**	25.33	16.90	**30.39**	**30.39**	24.23
(v)	50.79 ± 0.00	53.26	**76.22**	51.50	53.67	53.67	73.21
	1.37 ± 0.00	5.05	**42.33**	2.39	5.74	5.74	40.01

Table 4 Linear clustering rank in Table 2

Data	kmeans	kPC	PPC	TWSVC	RTWSVC	FRTWSVC	Ours
	AC	AC	AC	AC	AC	AC	AC
	MI	MI	MI	MI	MI	MI	MI
(a)	2	4	3	4	4	4	1
	1	4	3	4	4	4	2
(b)	4	5	3	2	5	5	1
	6	1	7	4	2	2	5
(c)	2	6	4	1	5	5	3
	1	7	6	3	4	4	2
(d)	3	5	1	2	6	7	4
	1	3	7	2	6	5	4
(e)	5	6	7	3	2	1	4
	5	6	7	2	3	1	4
(f)	1	3	6	2	3	3	7
	1	2	7	2	2	2	6
(g)	1	5	7	6	4	2	3
	1	6	7	3	5	2	4
(h)	2	7	6	5	1	3	4
	2	6	7	4	1	3	5
(i)	3	5	4	2	5	5	1
	3	5	4	2	6	6	1
(j)	6	7	5	4	2	3	1
	7	6	5	4	2	3	1
(k)	4	7	5	6	1	1	3
	5	7	6	3	1	1	4
(l)	2	7	6	5	3	3	1
	3	7	6	1	5	4	2
(m)	2	7	6	3	4	4	1
	2	6	7	3	4	4	1
(n)	5	3	7	6	1	2	3
	4	5	7	3	1	2	6
(o)	4	7	6	3	2	5	1
	3	6	5	6	2	4	1
(p)	3	4	1	7	4	4	1
	3	4	1	7	4	4	1
(q)	4	2	6	2	5	7	1
	1	3	3	3	6	7	2
(r)	4	7	2	4	6	3	1
	5	4	3	6	7	1	2
(s)	4	5	3	2	5	5	1
	6	1	6	5	1	1	4
(t)	1	5	3	4	5	5	2
	1	5	4	3	5	5	2
(u)	7	4	3	2	4	4	1
	4	5	3	2	5	5	1
(v)	3	4	2	5	5	5	1
	5	7	4	6	2	2	1
Mean$_{AC}$	3.27	5.23	4.36	3.64	3.73	3.91	**2.09**
Mean$_{MI}$	3.18	4.82	5.23	3.55	3.55	3.27	**2.77**

Table 5 Nonlinear clustering rank in Table 3

Data	kmeans	kPC	PPC	TWSVC	RTWSVC	FRTWSVC	Ours
	AC	AC	AC	AC	AC	AC	AC
	MI	MI	MI	MI	MI	MI	MI
(a)	6	3	1	7	3	3	2
	1	2	7	5	2	2	6
(b)	6	2	7	2	2	2	1
	6	2	7	2	2	2	1
(c)	6	4	7	1	4	3	2
	6	2	7	1	2	4	5
(d)	3	4	7	2	4	4	1
	4	1	6	5	1	1	7
(e)	6	2	7	2	2	2	1
	6	2	7	2	2	2	1
(f)	1	7	4	2	5	5	3
	2	6	7	1	4	4	3
(g)	2	5	7	3	4	5	1
	2	4	7	3	6	4	1
(h)	6	3	1	7	3	3	2
	6	3	1	7	3	3	2
(i)	7	2	6	2	2	2	1
	7	2	6	2	2	2	1
(j)	7	4	1	3	4	4	1
	4	5	2	3	5	5	1
(k)	2	5	3	3	5	5	1
	1	5	3	3	5	5	2
(l)	7	2	5	6	2	2	1
	5	2	6	7	2	2	1
(m)	3	5	4	1	5	5	2
	3	5	4	1	5	5	2
(n)	2	5	4	3	5	5	1
	5	2	7	6	2	2	1
(o)	7	4	2	3	4	4	1
	7	3	6	2	3	3	1
(p)	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
(q)	7	4	2	2	4	4	1
	2	5	3	3	5	5	1
(r)	3	4	1	4	4	4	2
	2	3	1	3	3	3	3
(s)	6	7	3	4	2	5	1
	6	7	2	4	3	5	1
(t)	4	5	3	7	1	5	1
	3	1	7	4	5	1	5
(u)	6	2	5	7	2	2	1
	6	1	4	7	1	1	5
(v)	7	5	1	6	3	3	2
	7	5	1	6	3	3	2
Mean$_{AC}$	4.77	3.86	3.73	3.55	3.23	3.55	**1.36**
Mean$_{MI}$	4.18	3.14	4.64	3.55	3.05	2.95	**2.41**



(a) Dermatology

(b) Ecoli

(c) Glass

(d) Iris

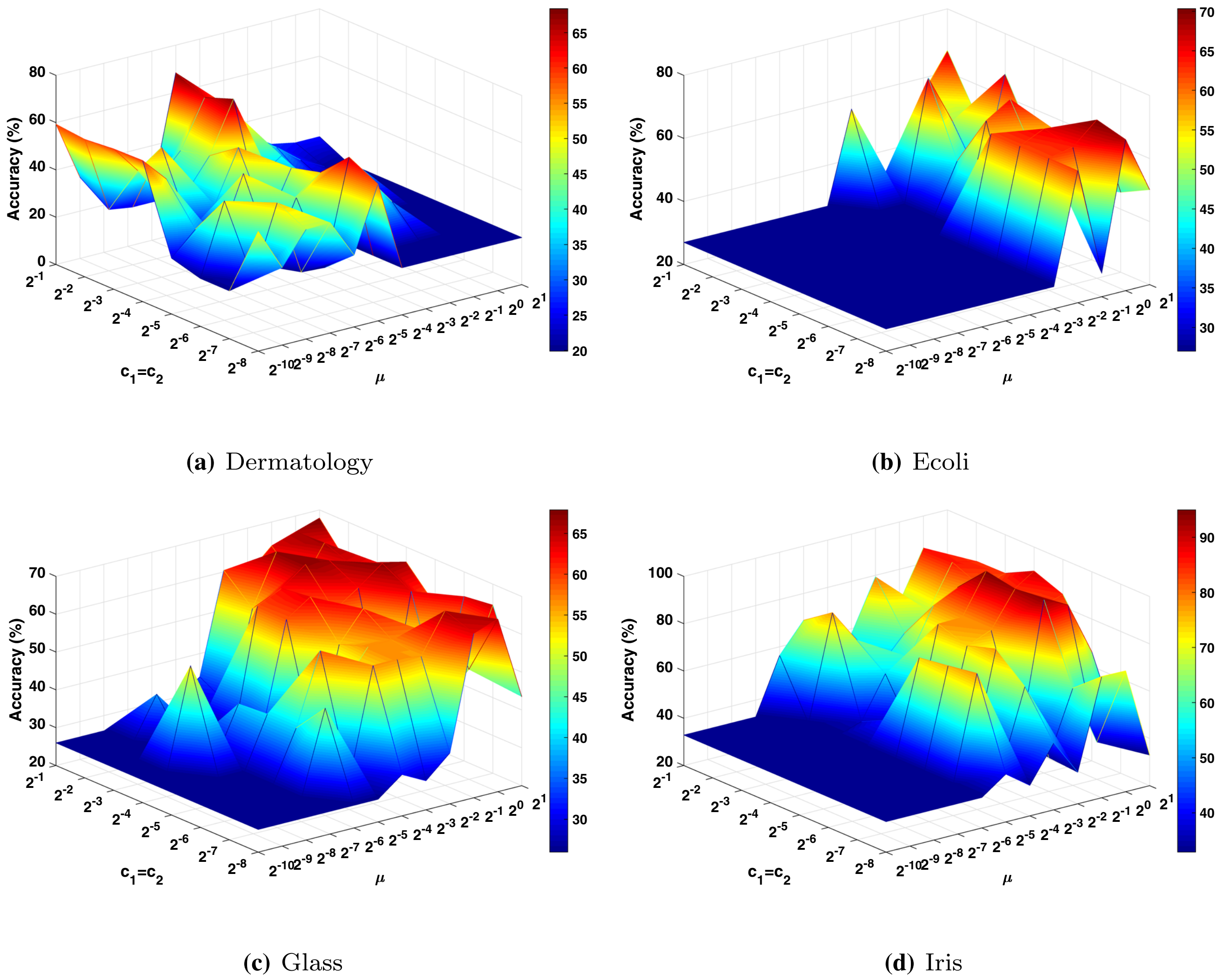
Fig. 2 Illustration of parameter influence of linear RampTWSVC on four datasets, where the parameters include c_1 and c_2

there are a finite number of ways to select p_1 and p_2, there are two finite numbers $0 < r_1 < r_2$ such that $(p_1^{(r_1)}, p_2^{(r_1)}) = (p_1^{(r_2)}, p_2^{(r_2)})$. Thus, we have $u_i^{(r_1)} = u_i^{(r_2)}$. That is to say, the objective values are equal in the r_1th and r_2th iterations. Since $p_1^\top Z_i^\top u_i \leq 0$ and $p_2^\top \hat{Z}_i^\top u_i \leq 0$ always hold, the objective value of (11) keeps nonincreasing in iteration. Therefore, the objective is invariant after the r_1th iteration, and then the algorithm would terminate at the r_1th iteration.

Let us consider the point $(u_i^{(r_1)}, p_1^{(r_1)}, p_2^{(r_1)})$. From the above proof, we have $G(u_i^{(r_1)}, p_1^{(r_1)}, p_2^{(r_1)}) = G(u_i^{(r_1)}, p_1^{(r_1+1)}, p_2^{(r_1+1)})$, where $G(\cdot)$ is the objective value of (11). If there is more than one global solution to (11) with fixed u_i, we always select the same one for the same u_i. Thus, we have $(p_1^{(r_1)}, p_2^{(r_1)}) = (p_1^{(r_1+1)}, p_2^{(r_1+1)})$, which indicates $(u_i^{(r_1)}, p_1^{(r_1)}, p_2^{(r_1)})$ is a local optimal point. □

4 Experimental results

In this section, we analyze the performance of our RampTWSVC compared with kmeans [10], kPC [14], PPC [15], TWSVC [17], RTWSVC [18] and FRTWSVC [18] on several benchmark datasets [31, 32]. All the methods were implemented by MATLAB 2017a on a PC with an Intel Core Duo Processor (double 4.2 GHz) with 16GB RAM. The parameters c in PPC, TWSVC, RTWSVC, FRTWSVC and c_1, c_2 in RampTWSVC were selected from $\{2^i | i = -8, -7, \ldots, -1\}$. For nonlinear case, the Gaussian kernel $K(x_1, x_2) = \exp\{-\mu ||x_1 - x_2||^2\}$ [28] was used, and its parameter μ was selected from $\{2^i | i = -10, -9, \ldots, 1\}$. The random initialization was used on kmeans, and the nearest neighbor graph (NNG) initialization [17] was used on other methods. In the experiments, we used the metric accuracy (AC) [17] and mutual information (MI) [33] to



(a) Dermatology

(b) Ecoli

(c) Glass

(d) Iris

Fig. 3 Illustration of parameter influence of nonlinear RampTWSVC on four datasets, where the parameters include c_1, c_2 and μ, and we set $c_1 = c_2$

measure the performance of these methods. We ran kmeans ten times on each dataset and recorded its mean value and standard deviation.

Table 1 shows the details of the benchmark datasets. Tables 2 and 3 exhibit the linear and nonlinear clustering results on the benchmark datasets, respectively. The highest metrics among these methods on each dataset are in bold. From Table 2, it can be seen that our linear RampTWSVC performs better than other linear methods on seven datasets in terms of both AC and MI, and it is more accurate than other methods on other six datasets. On the rest of nine datasets, our linear RampTWSVC is also competitive with the best one. From Table 3, it is obvious that our nonlinear RampTWSVC has much higher AC and MI over other methods on most of the datasets. More precisely, we reported the rank of each method on these datasets in Tables 4 and 5 from Tables 2 and 3, respectively, where the numbers represent the ranks of these seven methods, the same number implies the same performance and the smaller number implies the higher performance. The mean metrics were also reported at the last two columns in these tables. It is obvious that our RampTWSVC owns the smallest mean metrics among these methods, which indicates that our RampTWSVC performs better than other methods on these datasets. It is worth to notice that there are several datasets on which other methods perform better than RampTWSVC. For example, kmeans obtains a much higher performance than others on dataset (g) in Table 2, which implies the cluster centers in dataset (g) are more suitable for points than planes. In these plane-based methods, our RampTWSVC cut off the influence of the samples far from the cluster centers, and the bounds are decided by the parameters. So the parameter selection is very important in our RampTWSVC.

To evaluate the parameter influence in our RampTWSVC, as examples, we reported the metric ACs with different parameter settings on four datasets. Figure 2 exhibits the results of linear RampTWSVC with parameters c_1 and c_2. It can be seen that the above two parameters play an important role in this method. Thence, we should be very careful about the parameter selection. Figure 3 exhibits the results of nonlinear RampTWSVC with parameters c_1, c_2 and μ, where $c_1 = c_2$ for simplicity. From Fig. 3, it is obtained that we cannot obtain a desired result if μ is improper, which indicates the parameter μ is more important that the other two parameters for nonlinear RampTWSVC. This is obviously because the kernel mapping plays an more important role in the nonlinear case.

5 Conclusions

A plane-based clustering method (RampTWSVC) has been proposed in this paper. RampTWSVC hires the bounded ramp functions to reduce the influence of the samples far from the cluster centers, and it performs well in a robust manner. The nonconvex programming problems in RampTWSVC are recast to a series of mixed-integer programming problems and solved by a proposed alternating iteration algorithm. The local solutions of the nonconvex problems are guaranteed in theory. RampTWSVC contains both the linear and nonlinear formations. Experimental results on several benchmark datasets have indicated that our RampTWSVC performs much better than other plane-based clustering methods on many benchmark datasets. For practical convenience, the corresponding RampTWSVC MATLAB code has been uploaded upon http://www.optimal-group.org/Resources/Code/RampTWSVC.html. Future work includes the parameters regulation and efficient solver design for our RampTWSVC.

Acknowledgements This work is supported in part by National Natural Science Foundation of China (Nos. 61966024, 11501310, 61866010, 11871183 and 61703370), in part by Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (No. NJYT-19-B01), in part by Natural Science Foundation of Hainan Province (No. 118QN181), in part by Scientific Research Foundation of Hainan University (No. kyqd(sk)1804) and in part by Zhejiang Provincial Natural Science Foundation of China (No. LQ17F030003).

Compliance with ethical standards

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

References

1. Jain A, Murty M, Flynn P (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
2. Mehmood R, Zhang G, Bie R, Dawood H, Ahmad H (2016) Clustering by fast search and find of density peaks via heat diffusion. Neurocomputing 208:210–217
3. Zhou J, Chen L, Chen C, Zhang Y, Li H (2016) Fuzzy clustering with the entropy of attribute weights. Neurocomputing 198:125–134
4. Zhou Z (2017) A brief introduction to weakly supervised learning. Natl Sci Rev 5(1):44–53
5. Berry M (2004) Survey of text mining I: clustering, classification, and retrieval, vol 1. Springer, Berlin
6. Ilin R (2012) Unsupervised learning of categorical data with competing models. IEEE Trans Neural Netw Learn Syst 23(11):1726–1737
7. Zhang H, Chow T, Wu Q (2015) Organizing books and authors by multilayer som. IEEE Trans Neural Netw Learn Syst 27(12):2537–2550
8. Tan P, Steinbach M, Kumar V (2005) Introduction to data mining, 1st edn. Addison-Wesley Longman Publishing Co. Inc, Boston
9. Huang X, Ye Y, Zhang H (2014) Extensions of kmeans-type algorithms: a new clustering framework by integrating intracluster compactness and intercluster separation. IEEE Trans Neural Netw Learn Syst 25(8):1433–1446
10. Jain A, Dubes R (1988) Algorithms for clustering data. Prentice-Hall Inc, Upper Saddle River
11. Mangasarian O, Wild E (2006) Multisurface proximal support vector classification via generalize eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
12. Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
13. Wang Z, Shao Y, Bai L, Li C, Liu L, Deng N (2018) Insensitive stochastic gradient twin support vector machines for large scale problems. Inf Sci 462:114–131
14. Bradley P, Mangasarian O (2000) k-plane clustering. J Global Optim 16(1):23–32
15. Shao Y, Bai L, Wang Z, Hua X, Deng N (2013) Proximal plane clustering via eigenvalues. Proc Comput Sci 17:41–47
16. Shao Y, Guo Y, Wang Z, Yang Z, Deng N (2017) k-proximal plane clustering. Int J Mach Learn Cybernet 8(5):1537–1554
17. Wang Z, Shao Y, Bai L, Deng N (2015) Twin support vector machine for clustering. IEEE Trans Neural Netw Learn Syst 26(10):2583–2588
18. Ye Q, Zhao H, Li Z, Yang X, Gao S, Yin T, Ye N (2018) L1-norm distance minimization-based fast robust twin support vector k-plane clustering. IEEE Trans Neural Netw Learn Syst 29(9):4494–4503
19. Khemchandani R, Pal A, Chandra S (2018) Fuzzy least squares twin support vector clustering. Neural Comput Appl 29(2):553–563
20. Bai L, Shao Y, Wang Z, Li C (2019) Clustering by twin support vector machine and least square twin support vector classifier with uniform output coding. Knowl-Based Syst 163:227–240
21. Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
22. Collobert R, Sinz F, Weston J, Bottou L (2006) Large scale transductive SVMS. J Mach Learn Res 7:1687–1712
23. Cevikalp H (2017) Best fitting hyperplanes for classification. IEEE Trans Pattern Anal Mach Intell 39(6):1076–1088

24. Tian Y, Mirzabagheri M, Bamakan S, Wang H, Qu Q (2018) Ramp loss one-class support vector machine: a robust and effective approach to anomaly detection problems. Neurocomputing 310(8):223–235
25. Liu D, Shi Y, Tian Y, Huang X (2016) Ramp loss least squares support vector machine. J Comput Sci 14:61–68
26. Souvenir R, Pless R (2005) Manifold clustering. In: Tenth IEEE international conference on computer vision, ICCV 2005, vol 1, IEEE, pp 648–653
27. Cao W, Haralick R (2006) Nonlinear manifold clustering by dimensionality. In: 18th international conference on pattern recognition. ICPR 2006, vol 1. IEEE, pp 920–924
28. Khemchandani R, Chandra S (2009) Optimal kernel selection in twin support vector machines. Optim Lett 3:77–88
29. Platt J (1999) Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods-support vector learning. MIT Press, Cambridge, pp 185–208
30. Wang Z, Shao Y, Wu T (2013) A ga-based model selection for smooth twin parametric-margin support vector machine. Pattern Recogn 46(8):2267–2277
31. Blake C, Merz C (1998) UCI Repository for Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html
32. Li C, Shao Y, Yin W, Liu M (2019) Robust and sparse linear discriminant analysis via an alternating direction method of multipliers. IEEE Trans Neural Netw Learn Syst **(in press)**
33. Liu H, Fu Y (2015) Clustering with partition level side information. In: 2015 IEEE international conference on data mining (ICDM). IEEE, pp 877–882

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