Applied Intelligence (2023) 53:1254–1270
https://doi.org/10.1007/s10489-022-03493-5

Robust semi-supervised clustering via data transductive warping

Peng Zhou[1,2,3] · **Ni Wang**[1,2,3] · **Shu Zhao**[1,2,3] · **Yanping Zhang**[1,2,3]

Accepted: 10 March 2022 / Published online: 27 April 2022

Abstract
In practical applications, we are more likely to face semi-supervised data with a small amount of independent class label or constraint information and many unlabeled instances. For semi-supervised clustering, taking advantage of the small portion of preliminary label information can significantly improve the discriminability of representations. Spectral clustering has the benefits of handling any shape data distribution and converging to the optimal global solution but is susceptible to noisy data. However, it is inevitable to contain noise for real-world applications that significantly reduce clustering performance. Motivated by this, we propose a novel Robust Semi-supervised Spectral Clustering method (named RSSC) to address clustering on noise semi-supervised datasets. Specifically, in terms of data transductive warping, we map the entire semi-supervised dataset into a new data space where labeled data is close to the canonical coordinate system, and unlabeled data with similar characteristics should be close to those labeled data. The noise data is close to the origin of the coordinate and form the noise cluster because there is no guidance. Finally, samples in the same cluster are close, and different clusters are separated. Extensive experimental results on sixteen real-world datasets demonstrate that RSSC outperforms other state-of-the-art clustering methods on performance and robustness.

Keywords Clustering · Semi-supervised clustering · Noisy data · Spectral clustering · Robustness · Data transductive warping

1 Introduction

Data clustering is a core technology for machine learning and data mining, which aims to divide a set of unlabeled data into multiple clusters so that the data in the same clusters are more "similar" to each other [1]. Motivated by the fact that compared with the small number of labeled data in real-world applications, it is relatively easy to obtain a large amount of unlabeled data, semi-supervised learning was developed and applied widely [47]. Semi-supervised clustering employs a small amount of labeled data to aid unsupervised clustering and has widely applied in different practical applications, such as natural language processing [4], computer vision [34], bioinformatics [2], and image segmentation [38]. Generally speaking, there are two ways to use supervised information [6]. One is to obtain constraints based on the existing supervision information: must-link and the cannot-link constraint. The other is to use labeled data to assist clustering directly. For real-world applications, it is inevitable to contain noisy data that may significantly reduce the algorithm's performance. There are mainly two strategies for noise data clustering: (1) try to obtain the correct clustering from noisy data; (2) try to obtain the correct clustering by discriminating noise data. The former focuses on data partitioning without considering noise, and the output result is noisy clustering. The latter focuses on simultaneous clustering and denoising, and the output result is noise-free clustering. In reality, the latter is more challenging and meaningful. There are some works focus on robust data clustering, such as the algorithms

✉ Peng Zhou
doodzhou@ahu.edu.cn

Ni Wang
wangni613@hotmail.com

Shu Zhao
zhaoshuzs@ahu.edu.cn

Yanping Zhang
zhangyp2@gmail.com

[1] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui Province, 230601, P.R. China

[2] School of Computer Science and Technology, Anhui University, Hefei, Anhui Province 230601, P.R. China

[3] Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui Province 230601, P.R. China

based on density [42, 46], via node cutting [14], via rank minimization [35], or decomposing the similarity graph into two latent factors [7]. These robust models can tackle the issue of noises or outliers effectively in the unsupervised field.

Besides, some outlier detection methods have been proposed in recent years, such as [10, 23, 33]. Theoretically, it is possible to apply outlier detection before semi-supervised clustering. However, we generally do not do this. The main reason is that these two tasks are strongly coupled and should not be conducted as two separate tasks [19]. Besides, combining clustering and outlier detection can bring additional benefits include: (1) the resulting clusters tend to be compact and semantically coherent; (2) the clusters are more robust against data perturbations; and (3) the outliers are contextualized by the clusters and more interpretable [24]. Therefore, for robust semi-supervised clustering, we complete the noise detection and clustering simultaneously. In other words, we take the entire noise dataset as the target and consider the problem from a global perspective. The designed method should consider maximizing the use of a small amount of label information and complete the noise detection and clustering simultaneously.

Specifically, in some real-world clustering tasks, such as images and texts, the dimensionality of data can be extremely high. Therefore, there will inevitably be a lot of irrelevant and redundant features. If we cluster directly on the original datasets, these redundant and noisy information may degrade the performance and bring unnecessary computational costs. One common point of various methods for noise removal, model reduction, and feasibility reconstruction is to replace the original data with a low-dimensional representation obtained by subspace approximation. Therefore, to overcome the curse of dimensionality, some non-negative matrix factorization (NMF) and low-rank representation (LRR) based methods were proposed to obtain low-dimensional representation for noise data [9, 25, 40]. According to the experimental settings and results, all these aforementioned methods may have sound effects in semi-supervised robustness. However, there are two main shortcomings for these robust clustering methods: (1) most algorithms obtain the optimal solution after multiple iterations, and the locally optimal solution is iteratively reduced to get the final result, but this may not be optimal in global; (2) most algorithms add little noise in the experiments, and when the noise ratio is large, the clustering performance of these algorithms decrease significantly.

Based on the theory of spectrograms, spectral clustering has the advantage of being able to cluster in a sample space of any shape and converge to the optimal global solution. The essence of spectral clustering is to transform the clustering problem into the optimal division problem of graphs and has been widely used in text clustering [43], gene expression analysis [44], and other fields [13]. However, spectral clustering is susceptible to noisy data, while that is inevitable to contain noise in real-world applications. Therefore, we refine spectral clustering for robust semi-supervised clustering to achieve robustness and better clustering results in this paper. As far as we know, this is the first work to focus on the robustness of spectral clustering for semi-supervised noise data.

Transductive warping is the process of mapping the original data to a new space [18]. In general, assuming that X represents the original data, Y represents the data after the warping that we need to find. Constructing a loss function to obtain Y can be considered a transductive process called transductive warping. Transductive warping can be used in spectral clustering to reduce the influence of noise [18], which is created by affine transformation and elastic deformation of existing data. Each cluster of an arbitrary shape is mapped as a relatively compact cluster. The noise points are also mapped to form a relatively compact cluster, and different clusters become well separated. By applying data warping to reshape the noisy data, the block structure (destroyed by noise) of the affinity matrix can be recovered.

Semi-supervised clustering is more in line with actual application scenarios because labels are usually difficult or expensive to obtain. Spectral clustering has the advantages of handling any shape data distribution and converging to the optimal global solution but is susceptible to noisy data. However, it is inevitable to contain noise in real-world applications. Motivated by this, we propose a new Robust Semi-supervised Spectral Clustering algorithm for noise data, named RSSC. After data warping, the labeled data is close to the unit canonical coordinate system, while the unlabeled data is close to the labeled data with similar characteristics. The noise data will be close to the origin in canonical coordinates without label guidance. Thus, each cluster of an arbitrary shape is mapped to a relatively compact cluster, and different clusters are well separated. More details can be seen in Section 3.1. The contributions of this paper include the following:

- Inspired by the idea of data transductive warping, RSSC can restore the affinity matrix block structure destroyed by noise, which means that RSSC regards noisy data as an independent cluster. RSSC can simultaneously identify noise and complete data clustering tasks even for a large amount of noise.
- In RSSC, the class label information is converted into a matrix form and used to construct the objective function. The main object function we constructed is convex and can obtain the optimal global solution. Thus, RSSC is efficient and can improve the robust clustering performance significantly.

– Extensive experiments have been conducted on sixteen real-world datasets to verify the effectiveness and robustness of RSSC when compared with other state-of-art algorithms. Meanwhile, the robustness assessment experiments indicate that RSSC is very good at identifying noise data, even noise is considerable.

The rest of this paper is organized as follows. In Section 2, we describe the related work. In Section 3, we introduce our new robust semi-supervised clustering algorithm in detail. Section 4 verifies the superiority and effectiveness of the RSSC algorithm through the experiments on different types of datasets. The final part is the conclusions of this paper and some future research prospects.

2 Related work

Semi-supervised clustering is an important research direction in data mining, which uses limited prior knowledge to guide the search process and improve clustering quality. In recent years, some new semi-supervised clustering algorithms have been proposed. For example, Śmieja et al. [31] introduced a neural network framework for semi-supervised clustering with pairwise (must-link or cannot-link) constraints. Yu et al. [45] proposed a new double weighting semi-supervised ensemble clustering method based on selected constraint projection, which applies constraint weighting and ensemble member weighting to address the limitations. Ienco et al. [11] proposed a new semi-supervised clustering algorithm that directly exploits prior knowledge, under the form of labeled examples, avoiding the necessity to derive constraints. Wu et al. [41] took advantage of the supervised information by using them to construct a superior affinity matrix. Mai et al. [21] proposed a new spectral semi-supervised clustering method using the known data to complete good performance. However, all these semi-supervised models are not robust to noises, and it is inevitable to contain noise in real-world applications.

Besides, metric learning plays an essential role in many machine learning algorithms and is a fundamental problem in data mining and knowledge discovery [39]. More recently, researchers have given much attention to metric learning for semi-supervised algorithms. For instance, Baghshah et al. [5] considered the topological structure of data along with both positive and negative constraints, proposed a kernel-based metric learning method that provides a non-linear transformation. Li et al. [17] proposed a novel semi-supervised clustering approach based on deep metric learning, which leverages deep metric learning and semi-supervised learning effectively in a novel way. Sanodiya et al. [28] proposed a new kernel semi--supervised distance metric learning using a multi-objective optimization approach to overcome the problems associated with the K-means clustering algorithm. Shen et al. [29] proposed two types of distributed semi-supervised metric learning frameworks, which make use of both labeled and unlabeled data pairs. Nevertheless, these metric learning semi-supervised algorithms do not consider the effect of noise during clustering.

In order to solve the influence of noise on the clustering results, some robust clustering methods are proposed. More specifically, Zhou et al. [46] used KDE (kernel density estimation) to optimize local density and proposed a robust clustering algorithm named IVDPC, which solving the classification problem of data with different shapes and distribution. However, KDE is acceptably accurate in one-dimensional (1D) or 2D data, but becomes highly inaccurate for higher dimensional or sparse data. Bojchevski et al. [7] proposed the RSC that can enhance spectral clustering's robustness by decomposing the similarity graph into two latent factors: sparse corruption and clean data. Their experiments demonstrated the robustness of RSC against spectral clustering methods. However, RSC requires constant iterative calculations. Thus, an edge marked as corrupted in a previous iteration might be evaluated as non-corrupted later, which may decrease clustering performance. Tao et al. [35] proposed a novel Robust Spectral Ensemble Clustering (RSEC) algorithm, which not only targeted at a denoising task for the co-association matrix but also focused on revealing its cluster structure. Nevertheless, RSEC needs to solve n optimization subproblems over n data points and calculate the graph Laplacian matrix's eigenvectors through the Singular Value Thresholding (SVT) operator, which should take much time.

Due to the advantages of semi-supervised clustering and considering the inevitable noise problem, robust semi-supervised clustering algorithms are gradually being studied by scholars. For example, Wang et al. [40] used the non-negative matrix factorization method(NMF), which can employ the label information with a constraint matrix and address the noisy and sparse data simultaneously. However, graph dual regularized NMF learning algorithms may have divergent points, limiting the performance when low-dimensional representations are used for clustering. The same problem exists in several other non-negative matrices factorization-based methods, such as [9, 25]. Peng et al. [25] utilized the dual semi-supervised information to learn the more discriminative data representation, simultaneously adopted correntropy as the similarity measure for reducing the negative influence of non-Gaussian noise and outliers. However, this method cannot obtain an excellent local optimal solution and only verify algorithm performance on image data. Fang et al. [9] proposed a robust semi-supervised subspace clustering method based on non-

negative low-rank representation to obtain discriminant Low-rank representation (LRR) coefficients, which can address the overall optimum problem by integrating the affinity matrix construction and subspace clustering. Nevertheless, it used the singular value decomposition (SVD), which is time-consuming if many samples are vast. Lai et al. [16] proposed a new framework that generates base partitions in an unsupervised manner and attributes different weights to each cluster, which using k-means with both random sampling and random subspace techniques. Nevertheless, it also has obvious limitations: the algorithm must determine each cluster's center in advance. The choice of these cluster centers determines the quality of the clustering results. Besides, the algorithm is only effective for a small amount of noise, and it can only deal with low-noise data (up to 20% of noise).

Based on the transductive theory, the data warping method is applied in different areas. For example, Ma et al. [20] proposed kernel warping to induce feature representations that respect invariances that reach beyond transformation. The framework is efficient and flexible when applied to the convolutional kernel network. Qian et al. [26] incorporated the underlying manifold structure of both labeled and unlabeled data into the learning of a classifier via warping a reproducing kernel Hilbert space for sensor-based activity recognition problems. Ionescu et al. [12] derived an approximate learning procedure for data-dependent kernels that performs well in practice, which relies on low-dimensional kernel approximations and a warping term depending on a geometric operator. All these methods mentioned above demonstrate the effectiveness of data warping.

Inspired by the idea of data transductive warping, we propose a new robust semi-supervised spectral clustering algorithm for noise data, named RSSC, which integrates labeled instances as necessary information to improve learning accuracy. RSSC can obtain a globally optimal solution and perform well even with considerable noise.

3 Robust semi-supervised spectral clustering

This section first gives the formal definition of the research problem and our new algorithm's main ideas. Then, we present our primary constructed loss function and its optimization in detail. Based on these, we propose the novel robust semi-supervised spectral clustering algorithm (RSSC) and discuss it in detail.

3.1 The problem definition and main ideas

Given a noisy dataset with n data points $X = \{x_1, x_2, ..., x_n\}$, $x_i \in \mathbb{R}^d$, $i = 1, ..., n$, where subset S ($S \subset X$) represents a small number of labeled information in it. For semi-supervised clustering, we aim to learn the clusters $C_1, ..., C_k$ from X that is "as good as possible".

Fig. 1 illustrates the basic idea of data transductive warping. To facilitate observation, we simplified the canonical coordinate system, assuming that $I(i,\cdot)^T$ forms the canonical coordinate system of $\mathbb{R}^n$, where $I(i,\cdot)$ denotes the i^{th} row vector of the identity matrix I. In Fig. 1(a), "△" and "◯" represent two different classes of these data and the characteristics of the instances that belong to the same class are similar. Meanwhile, "☆" represents some noise instances among them. As shown in Fig. 1(b), after transductive warping, instances belonging to the differnt classes are separated. As mentioned above, transductive warping is the process of mapping the original data to a new space [18]. In general, assuming that X represents the original data, Y represents the data after the warping that we need to find. Constructing a loss function to obtain Y can be considered a transductive process, called transductive warping. So, in the new data space, the instances belonging to the same class are mapped to a relatively compact cluster, and the noise points are also mapped to a relatively compact cluster close to the origin in canonical coordinate. Therefore, we can cluster the instances and distinct noise points simultaneously after data transductive warping.

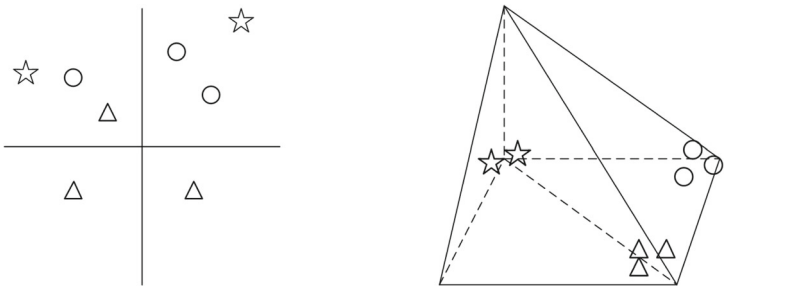
[Problem Definition] Inspired by the idea of data warping [18] and the theory of transductive inference [37], we construct the objective function for robust semi-supervised clustering as follows:

$$Z^* = \arg\min_{Z} \mathbf{\Omega}(Z), \tag{1}$$

$$\mathbf{\Omega}(Z) = \|Z_l - I\|^2 + \|Z_u - Z_l\|^2 + \frac{\mu}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{z_i}{\sqrt{d_{ii}}} - \frac{z_j}{\sqrt{d_{jj}}}\right)^2, \tag{2}$$

where $Z = (Z_l, Z_u) \in \mathbb{R}^n$ is the dataset after mapping, Z_l and Z_u are labeled and unlabeled data in the new space, μ is the regularization parameter which controls the balance of the loss function, $W = [w_{ij}]_{n\times n}$ represents the symmetric adjacency matrix connecting the nodes x_i and x_j according to (3), $d_{ii} = \sum_j w_{ij}$ and $d_{jj} = \sum_i w_{ij}$.

Specifically, the first item of (2) aims to make the labeled instances in the new data space as close to I as possible. The original data X is warped into Z, where $Z = (Z_l, Z_u) \in \mathbb{R}^{n\times n}$, $Z_l, Z_u \in \mathbb{R}^{n\times n}$. For Z_l, if x_i is unlabeled, $Z_l^i = (0, 0, ..., 0)^T$. For Z_u, if x_i is labeled, $Z_u^i = (0, 0, ..., 0)^T$. In other words, for Z_l, we only consider labeled data, while Z_u only considering unlabeled data. Besides, x_l is converted into z_l in the new space near $I(i,\cdot)^T$, and x_u is converted into z_u near z_l with similar features, where x_l and x_u represent original labeled and unlabeled data respectively. $I(i,\cdot)$ denotes the i^{th} row vector of identity matrix I, and $I(i,\cdot)^T$, $i = 1, ..., n$, form the canonical coordinate system of $\mathbb{R}^n$. In other words, we aim to map the original data to the canonical coordinate system through the data warp,



(a) original instances distribution (b) new spatial instances distribution after transductive warping

Fig. 1 Illustration of the basic idea of data transductive warping, where different shapes denote differnt classes. Compared with the distribution of the original instances (a), after data transductive warping, instances belonging to the differnt classes are separated (b)

making it possible to distinguish between clusters better. The establishment of this canonical coordinate system, due to the small amount of label data, should be constructed with these labeled instances. Therefore, $\|Z_l - I\|^2$ is equivalent to creating new space data with a spatial limit, and the new data can only be in the canonical coordinate system. After transforming and mapping X_l to Z_l, based on the idea that the characteristics of points in the same cluster are similar, the second item of (2) makes the unlabeled data Z_u that in the same cluster should be closer to Z_l. Thus, combining the first and second items, we can make the unlabeled instances in the same cluster more compact with the labeled instances in the new data space. The last item of (2) is the penalty item. It ensures that each component of the points sharing the same cluster has a relatively large and similar value. In contrast, the component of the points should have different values and be close to zero for the other clusters. The details can be seen in Section 3.2.

In general, semi-supervised transductive learning can improve unsupervised learning tasks by the small amount of labeled data. When be applied in semi-supervised clustering, the prior label information can better guide the clustering process. Specifically, the new dataset after data transductive warping is divided into several clusters according to a specific inter-sample measurement criterion. The samples in the same cluster are more similar under the current measurement criterion, and the samples in different clusters should be different. In Semi-supervised learning [36], transductive methods do not construct a classifier for the entire input space, and their predictive power is limited to exactly those objects that it encounters during the training phase. Therefore, transductive methods have no distinct training and testing phases. Since no model of the input space exists in transductive learners, information has to be propagated via direct connections between data points.

Fig. 2 shows the flow charts of RSSC and standard spectral clustering. The difference between RSSC and standard spectral clustering is marked with a red dashed box in Fig. 2(a). Specifically, standard spectral clustering works directly on the original noise dataset without considering the influence of noise on the similarity matrix. RSSC first constructs the overall similarity matrix with p-nearest neighbors to obtain the normalized Laplacian matrix, and then together with the label information, the data in the new data space is obtained through data warping. In the new data space, the instances in the same cluster are relatively compact, the instances in different clusters are relatively separated, and the noise is regarded as a new cluster. Finally, we apply spectral embedding clustering in the new data space. RSSC consists of four main components:(1) generate the p-nearest neighbor similarity matrix from the original dataset, and then calculate the normalized graph Laplacian matrix; (2) the label information is combined with the normalized graph Laplacian matrix and applied to the objective function to obtain a new dataset of the canonical coordinate system; (3) recalculate the p-nearest neighbor similarity matrix and normalized Laplacian matrix for the data in the new space; (4) apply the normalized spectral clustering method on the dataset of the new data space. The experimental results in Section 4 indicate the effectiveness and robustness of RSSC.

3.2 The solution of the objective function

Suppose $W = [w_{ij}]_{n\times n}$ represents the symmetric adjacency matrix connecting the nodes x_i and x_j. Most existing spectral clustering works [15, 18] construct fully connected similarity graphs to form the similarity matrix. However, this increases the complexity of the calculation, and one has to pay attention to selecting a suitable scaling factor. Thus, this paper uses a p-nearest neighbor graph to make the formed matrix sparse. Specifically, we use an symmetric p-nearest neighbor graph to construct a similarity graph W, where $kNN(x_i)$ represents the points of p domains of x_i. We use the following formula to construct the symmetric graph:

$$w_{ij} = w_{ji} = \begin{cases} 0, & x_i \notin kNN(x_j) \wedge x_j \notin kNN(x_i) \\ 1, & x_i \in kNN(x_j) \vee x_j \in kNN(x_i) \end{cases} \tag{3}$$

Inspired by the idea of data warping in [18], we aim to use labeled data to guide clustering so that the data in the same cluster after data warping should be as close as possible.

For the last item of (2), according to the Laplace operation on the graph [32], the following derivation can be made:

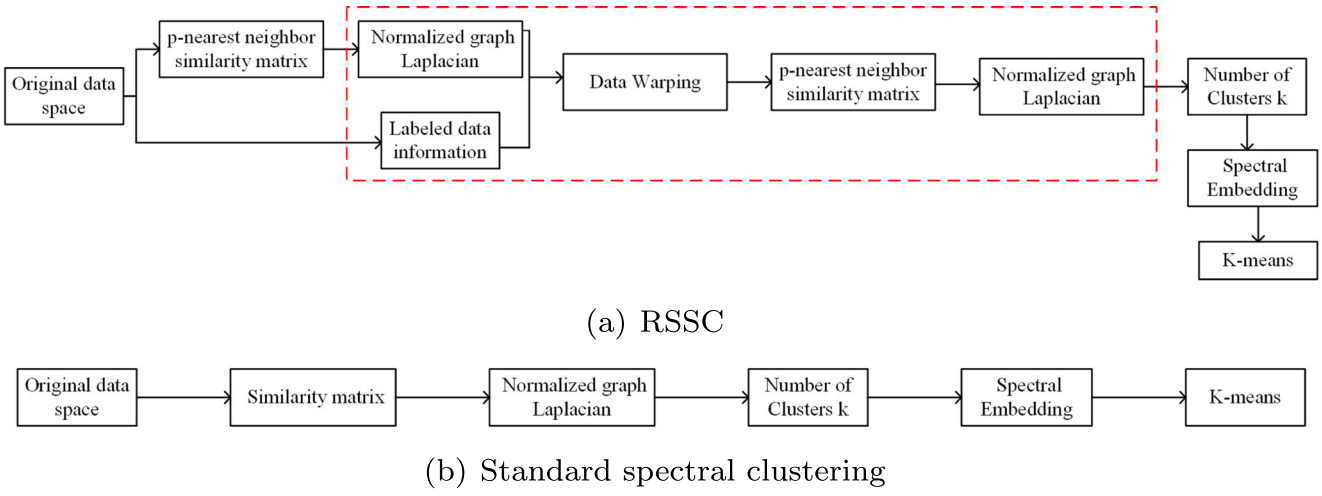


Fig. 2 The flow charts of RSSC(a) and standard spectral clustering(b). The difference between RSSC and standard spectral clustering is marked with a red dashed box

$$\begin{aligned}
&\mu \sum_{i,j=1}^{n} w_{ij}\left(\frac{z_i}{\sqrt{d_{ii}}} - \frac{z_j}{\sqrt{d_{jj}}}\right)^2 \\
&= \mu Z^T (I - D^{-1/2} W D^{-1/2}) Z \\
&= \mu Z^T D^{-1/2} (D - W) D^{-1/2} Z \\
&= \mu Z^T \bar{L} Z.
\end{aligned} \tag{4}$$

So, the loss function (2) can be further written as:

$$\mathbf{\Omega}(Z) = \|Z_l - I\|^2 + \|Z_u - Z_l\|^2 + \mu Z^T \bar{L} Z. \tag{5}$$

where the unlabeled data in Z_l are set to 0, and the labeled data in Z_u are set to 0 too. Then the first two items in (5) can be further equivalent to:

$$\begin{aligned}
&\|Z_l - I\|^2 + \|Z_u - Z_l\|^2 \\
&= \|SZ - I\|^2 + \|(I - S)Z - SZ\|^2 \\
&= (SZ - I)^T (SZ - I) + (Z - 2SZ)^T (Z - 2SZ) \\
&= \|Z - SI\|^2 + Z^T S Z + I - S
\end{aligned} \tag{6}$$

where $S \in \mathbb{R}^{n \times n}$ is the diagonal matrix given by $S_{ii} = I_{[l]}(i)$. If x_i is unlabeled, the corresponding $s_i = 0$. Because the normalized cut [30] employ the smallest eigenvector of the graph Laplacian, the optimization $\mu Z^T \bar{L} Z$ of the last term of (5) can be written as $\mu tr(Z^T K^{-1} Z)$, where K^{-1} represents the inverse of the kernel when K is a nonsingular matrix. If K is a singular matrix, K^{-1} represents a pseudo-inverse. With the help of graph kernel, particularly, if $r(\lambda) = \lambda$, $r(\bar{L})$ turns out to be the normalized graph Laplacian $\bar{L}$ and $K = \bar{L}^{-1}$, where λ is the eigenvalue of $\bar{L}$, $r(\lambda)$ is a general regularization operators generated on $\bar{L}$ according to [32, 48]. We use this kind of graph kernel in this paper.

In terms of (5) and (6), our objective function (2) can be rewritten as:

$$\mathbf{\Omega}^*(Z) = \|Z - SI\|^2 + Z^T S Z + I - S + \mu tr(Z^T K^{-1} Z). \tag{7}$$

We want the gap between the labeled data and the canonical coordinate system is as small as possible, and the data in the same class are as compact as possible. Thus, the final goal of (7) is:

$$Z^* = \arg\min_{Z} \mathbf{\Omega}^*(Z). \tag{8}$$

3.3 Optimization of problem

The second derivative of (7) to Z is

$$\frac{\partial \frac{\partial \mathbf{\Omega}^*(Z)}{\partial Z}}{\partial Z} = 2I + 2S + 2\mu K^{-1}. \tag{9}$$

Because $K^{-1} = \bar{L}$ is a positive definite matrix, $det(2I + 2S + 2\mu K^{-1}) > 0$, indicating that (7) is a strictly convex function. Thus, there is only one global minimum value. Taking the derivative of $\mathbf{\Omega}^*(Z)$ with respect to Z and setting it to zero yields:

$$\frac{\partial \mathbf{\Omega}^*(Z)}{\partial Z} = 2(Z - SI) + 2SZ + 2\mu K^{-1} Z = 0. \tag{10}$$

(10) can further simplified to $(I + S + \mu K^{-1}) = SI$. Thus, we can get the minimum solution of $\mathbf{\Omega}^*(Z)$ as:

$$Z^* = (I + S + \mu K^{-1})^{-1}(SI). \tag{11}$$

In order to facilitate data processing and accelerate the speed of seeking the optimal solution, we convert the result obtained by (11) to $Z^* \rightarrow \bar{Z}^*$, where $\bar{Z}^*$ is Z^* through linear scaling to [0,1]. The entire data transductive warping can be expressed as $\mathbf{\Theta}_K : X \rightarrow \mathbb{R}^n, x_i \mapsto \bar{Z}^*(i, \cdot)^T$, for $i = 1, ..., n$, $\bar{Z}^*(i, \cdot)^T$ refers to the i^{th} row vector of $\bar{Z}^*$.

In the transductive warping, we use the graph kernel principle on the Hilbert space. The advantages include: (1) it is suitable for high-dimension space, and (2) there is no need to reduce the data dimension that may lead to information loss. Then, the data in the new space can be clustered according to spectral clustering.

We further use spectral clustering to reduce the dimensionality of the data from $\mathbb{R}^n$ to a lower-dimensional space, which is equal to the number of clusters (including noise clusters). Assuming there are k clusters, we need to convert $\bar{Z}^* \rightarrow \mathbb{R}^k$. Let $\hat{L}$ represent the normalized

graph Laplacian in the $\mathbb{R}^n$ space, then the eigensystem of $\hat{L}$ is $\{(t_i, \zeta_i)|i = 1, ..., n\}$, where $\zeta_1 \leq ... \leq \zeta_n$, $T = [t_1, ..., t_k] \in \mathbb{R}^{n\times k}$ is the first k smallest eigenvectors of $\hat{L}$, each row of T is normalized to get $\bar{T} = [\bar{t}_{ij}]_{n\times k}$, where $\bar{t}_{ij} = t_{ij}/(\sum_{j=1}^{k} t_{ij}^2)^{1/2}$, $1 \leq i \leq n$. Let $\hat{y}_i \in \mathbb{R}^k$ be the vector corresponding to the i^{th} row of $\bar{T}$, then $\Phi_{\hat{L}} : \bar{Z}^* \to \mathbb{R}^k$, $\bar{Z}^*(i, \cdot)^T \mapsto \hat{y}_i$.

Based on these, we develop a new algorithm for robust semi-supervised spectral clustering, shown as Algorithm 1. It formally describes the use of supervised information on semi-supervised data to complete robust spectral clustering through data transductive warping. Steps 1-2 and Steps 5-6 use p-nearest neighbor to get the normalized graph Laplacian. The former aims to construct the graph kernel K, and the latter is to complete spectral clustering. Step 4 shows the data $\bar{Z}^*$ in the new space. Steps 7-8 get the eigenvectors of the first k normalized graph Laplacians $\hat{L}$ and normalize them. Step 9 is to perform k-means clustering on the eigenvectors.

Algorithm 1 Robust semi-supervised spectral clustering (RSSC).

Input: Nosiy dataset X, the matrix S of labeled information, number of clusters k, parameter p for p-nearest similarity matrix, parameter μ for loss function.

Output: Clustering $C_1, ..., C_k$.

1: Construct a symmetric p-nearest neighbor similarity matrix $W = [w_{ij}]_{n\times n}$ on X according to (3);
2: Construct the diagonal matrix $D = [d_{ij}]_{n\times n}$ on X, $d_{ii} = \sum_j w_{ij}$, and form the normalized graph Laplacian $\bar{L} = I - D^{-1/2} W D^{-1/2}$;
3: Calculate graph kernel $K = \bar{L}^{-1}$;
4: Calculate the transductive warping dataset Z^* according to $Z^* = (I + S + \mu K^{-1})^{-1}(SI)$, then linearly scale to [0,1] to get $\bar{Z}^*$;
5: Construct a symmetric p-nearest neighbor similarity matrix $\hat{W} = [\hat{w_{ij}}]_{n\times n}$ on $\bar{Z}^*$ according to (3);
6: Construct the diagonal matrix $\hat{D} = [\hat{d_{ij}}]_{n\times n}$ on $\bar{Z}^*$, $\hat{d_{ii}} = \sum_j \hat{w_{ij}}$, and form the normalized graph Laplacian $\hat{L} = I - \hat{D}^{-1/2} \hat{W} \hat{D}^{-1/2}$;
7: Calculate the first k smallest eigenvectors $[t_1, ..., t_k]$ of $\hat{L}$ and use them as column vectors to form a matrix $T \in \mathbb{R}^{n\times k}$;
8: Form the matrix $\bar{T} = [\bar{t}_{ij}]_{n\times k}$ from T by normalized to have unit length, that is set $\bar{t}_{ij} = t_{ij}/(\sum_{j=1}^{k} t_{ij}^2)^{1/2}$, $1 \leq i \leq n$. Let $\hat{y}_i \in \mathbb{R}^k$ be the vector corresponding to the i^{th} row of $\bar{T}$;
9: Cluster the points $(\hat{y}_i)_{(i=1,...,n)}$ with the k-means algorithm.

3.4 Algorithmic discussion

On optimal solution, the second derivative of (7) is

$$\frac{\partial \frac{\partial \mathbf{\Omega}^*(Z)}{\partial Z}}{\partial Z} = 2I + 2S + 2\mu K^{-1}. \tag{12}$$

For $K^{-1} = \bar{L}$ is a positive definite matrix, so $det(2I + 2S + 2\mu K^{-1}) > 0$. It means (7) is strictly convex for Z, and there is only one global optimal solution. Thus, our algorithm can converge to a global minimum, and the calculation is very efficient.

On required parameters analysis, there are three parameters in Algorithm 1. For clusters k, we set it to the number of clusters in clean data plus noise clusters. Parameter μ, which controls the trade-off of the entire loss function (7), is a positive regularization parameter for our new algorithm. Parameter p, which determines the number of neighbors, the optimal value is different for different datasets. More details of the analysis of these parameters will be explained in Section 4.3.

On the time complexity of Algorithm 1, the most time-consuming steps are Step 4 and Step 7. Specifically, step 4 finds the inverse matrix, and the time complexity is $O(n^3)$. Step 7 calculates the eigenvector of the matrix, and the time complexity is also $O(n^3)$. Therefore, the time complexity of RSSC is $O(n^3)$.

4 Experiments

In this section, we first introduce the experimental setup and evaluation metrics. Then, we analyze the parameters and the robustness under different noise ratios. Next, we compare RSSC with several state-of-the-art methods on sixteen different types of datasets. Meanwhile, we use some graphs to illustrate the effect of RSSC and its competing algorithms visually. Finally, we present the statistical testing on the comparison results.

4.1 Experimental setup

Datasets descriptions. We conducted experiments on ten UCI datasets ,two image datasets (USPS and MNIST) and two text datasets (Re0 and Oh0) [27], where datasets Vertebral Column, Parkinson's Disease Classification, and Pendigits-test are abbreviated as VC, PDC, and Pt, respectively. Simulating noise data on image and text data sets is described as follows:

– USPS-01: randomly select 500 instances from the numbers "0" and "1" as the clean dataset, and randomly choose 240 instances from digits "2"-"9" (30 per digit) as the noise data.

- MNIST-0: randomly select 900 instances from the number "0" in MMIST-10K as the clean dataset, and randomly choose 270 instances from "1"-"9" (30 for each) as the noise data.
- MNIST-012: randomly select 500 instances from numbers "0"-"2" in MMIST-10K as the clean dataset, and randomly choose 210 instances from "3"-"9" (30 for each) as the noise data.
- MNIST-0123: randomly select 500 instances from numbers "0"-"3" in MMIST-10K as the clean dataset, and randomly choose 300 instances from "4"-"9" (50 for each) as the noise data.
- Re0: select the instances labeled "interest", "trade", and "money" from Re0 as the clean dataset, and randomly select 100 instances (10 for each rest label) as the noise data.
- Oh0: select the instances labeled "Ethics", "Fundus-Oculi", "England" and "Heart-Valve-Prosthesis" from Oh0 as the clean dataset, and randomly select 120 instances (20 for each rest label) as the noise.

Table 1 shows the details of these datasets used in the experiments.

Competing algorithms. To verify the robustness of our new proposed algorithm, we compare RSSC with the following algorithms:

- **WECR k-means** [16]: a method that uses weighted consensus of random k-Means ensemble to construct an adaptive robust semi-supervised clustering framework.
- **RSEC** [35]: a low-rank representation (LRR) based method to perform spectral clustering and can deal with noise.
- **SSSC** [21]: a semi-supervised spectral clustering method which is consistent with both labelled and unlabelled data.
- **PCPSNMF** [41]: a symmetric NMF-based semi-supervised clustering method that uses supervised information to construct a superior affinity matrix.
- **RSC** [7]: this method uses sparse and latent decomposition of the similarity graph in spectral clustering to adapt to noisy data.
- **SC,k-means**: the basic Spectral Clustering and k-means algorithms.

Table 1 Real-world datasets

Index	Datasets	Samples	Features	Clusters
1	iris	150	4	3
2	wine	178	13	3
3	plrx	182	13	2
4	seeds	210	7	3
5	VC	310	6	3
6	wdbc	569	32	2
7	PDC	756	754	2
8	banknote	1372	5	2
9	yeast	1484	8	10
10	Pt	3498	16	10
11	USPS-01	1240	256	3
12	MNIST-0	1170	784	2
13	MNIST-012	1710	784	4
14	MNIST-0123	2300	784	5
15	Re0	1246	2886	4
16	Oh0	746	3182	5

Algorithm settings. All methods are executed in the PYTHON environment. SSSC, PCPSNMF, and RSEC are implemented by ourselves since the original authors did not provide the PYTHON source codes. All the experiments are repeated ten times to obtain the mean and standard deviation of clustering results. In the whole experiment, 10% of the instances are randomly selected as the labeled data. Meanwhile, we added 40% uniformly distributed noise into the UCI datasets to simulate the target noise datasets. We generate $0.4*\tilde{n}$ uniformly distributed noises in the interval $[a, b]$, where a, b means the noise generated in this interval covers as clean data as possible, and $\tilde{n}$ represents the number of instances in the original dataset.

All the parameter settings of comparison algorithms are consistent with the values given in the original papers. For WECR k-means, 100 base partitions are generated for each cluster ensemble, the hyperparameter γ is tuned in $[0, 1]$ with 0.1 as the step size, and generate $0.5N$ constraints (N is the number of instances). For RSEC, we set $r = 100$ as the basic partitions, $\lambda_1 = 0.1$, $\lambda_2 = 0.01$, $\rho = 1.3$, $\mu_{max} = 10^{10}$, $\epsilon = 10^{-7}$ as the values used in [35]. For SSSC, similarity matrices w_{ij} are computed with a Gaussian kernel function: $w_{ij} = exp(\|x_i - x_j\|^2/p)$, where p is the dimension of data vectors. For PCPSNMF, we set tradeoff parameter $\mu = 0.5$, penalty parameter $\alpha = 1$, maximum iteration number $Iter = 500$, 10% label information for each class was randomly chosen to generate constraints as supervised information. In the p-nearest neighbors similarity matrices, we set $p = 4$, $\sigma = 2$. For RSC, we construct nearest neighbor graphs with $k = 15$ neighbors, allowing half of the edges $m = 0.5$ to be removed per node, and set iteration number equals 50. For SC, use the method that is implemented in the sklearn library. In RSSC, two parameters are involved, according to the discussion in Section 4.3, the value of p is in the range of [7, 15] or [30, 35], $\mu = 50$. For each method and dataset,

we run the experiment 10 times and reported average and standard deviation results in the experiments.

4.2 Evaluation metrics

There are many clustering indicators, such as NMI, ACC, FMI, DI, and DBI [3, 22]. In this paper, we use Normalized Mutual Information (NMI) and Average Clustering Accuracy (ACC) for two reasons: (1) both metrics are the most common and frequently used clustering indicators; (2) for the datasets we used in experiments, the true labels are known in advance. Thus, with these two metrics, the performance of the clustering algorithm can be more clearly reflected. The range of NMI and ACC is [0,1].

NMI measures the mutual information entropy between resulted labels and the ground-truth labels. Given two sets of clusters H and $\overline{H}$,

$$NMI = \frac{\sum_{h\in H,\overline{h}\in\overline{H}} p(h,\overline{h})log(\frac{p(h,\overline{h})}{p(h)p(\overline{h})})}{\sqrt{\sum_{h\in H} p(h)log(\frac{p(h)}{n})}\sqrt{\sum_{\overline{h}\in\overline{H}} p(\overline{h})log(\frac{p(\overline{h})}{n})}}, \tag{13}$$

where $p(h)$ and $p(\overline{h})$ represent the marginal probability distribution functions of H and $\overline{H}$, respectively, induced from the joint distribution $p(h,\overline{h})$ of H and $\overline{H}$, and n is the number of samples.

ACC discovers the one-to-one relationship between clusters and classes.

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(l_i))}{n}, \tag{14}$$

where l_i and y_i are the clustering result and the ground truth cluster label, respectively, and n is the number of samples. $\delta(a,b)$ equals one if and only if $a = b$, otherwise it is zero. $map()$ is the permutation mapping function that maps each cluster index to a true class label.

4.3 Parameter analysis

In RSSC, two parameters are involved: (1) p, which determines the number of neighbors; (2) μ, which controls the trade-off of the entire loss function.

Since p is the number of neighbors of a sample, the feasible value of p should be different for different datasets. According to Table 1, we test the range of p from 5 to 100 with steps 3. μ is the regularization parameter and controls the balance of loss function (7). Thus, $\mu > 0$, and the value of μ should not be too large. We test the range of μ from 0 to 200 with steps 5. Due to space constraints, Fig. 3 only depicts the effect of these two parameters on the clustering performance (ACC and NMI) on datasets iris, seeds, and VC, where "mu" represents the parameter μ.

As shown in Fig. 3, for a fixed value of p, the fluctuations of ACC and NMI on these three datasets are small in cases of different values of μ. Especially on dataset seed, with the same value of p, different values of μ almost achieve the same performance. Thus, μ is not very sensitive for RSSC. In the following experiments, we set $\mu = 50$ as an experience value. With a fixed value of μ, different values of p greatly influence the performance of ACC and NMI. For different datasets, the number of instances varies from each other. Thus, it is not easy to specify a fixed value of p optimal for all different datasets. In general, a smaller value of $p([7, 35])$ can achieve better performance on these three datasets. Besides, we have done parameter analysis on all the other datasets to observe the optimal range of parameter p. In total, RSSC can perform better when the value of p is in the range of [7, 15] or [30, 35].

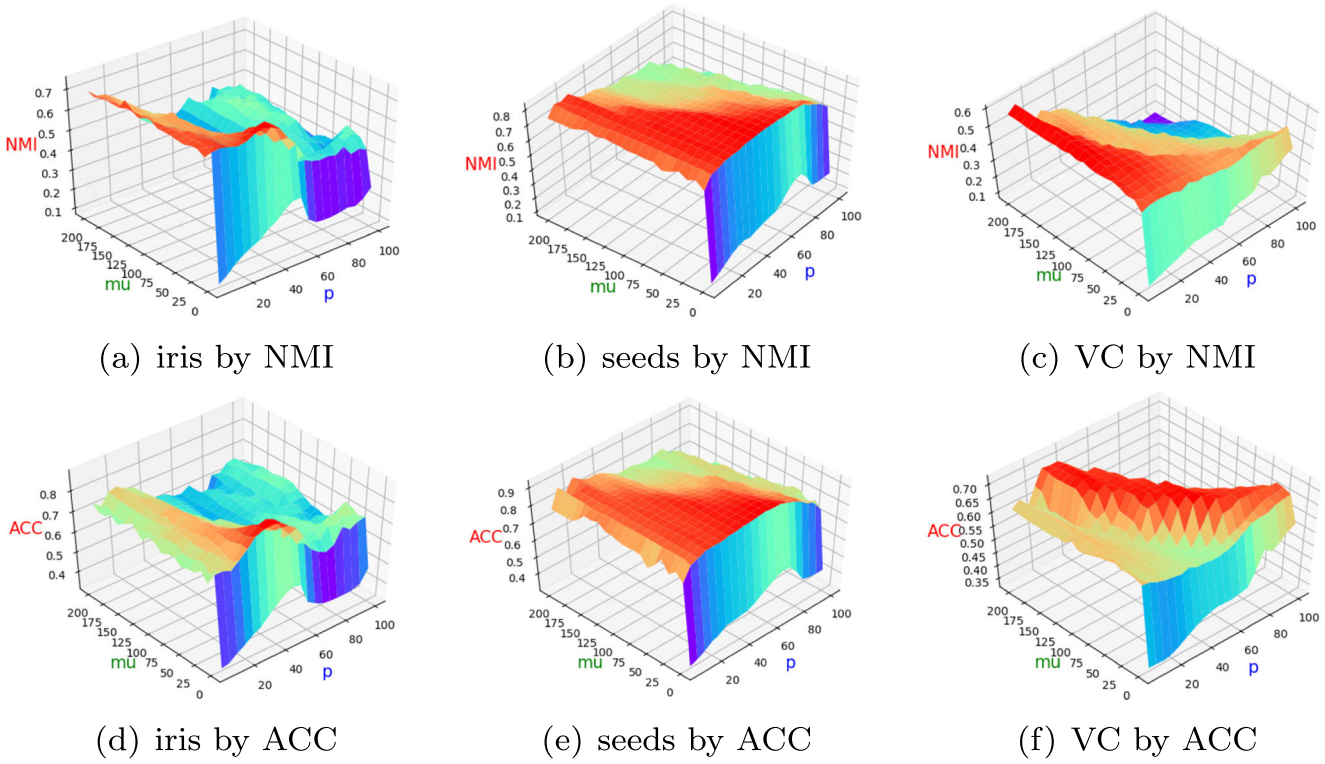
4.4 Robustness assessment

In this part, we verify the robustness of RSSC by adding different proportions of noise into real-world UCI datasets and applying all the competing algorithms on these new noisy datasets. If the noise is far away from the clean data, such noise is easy to be detected. Thus, we add noise that surrounds the clean data and covers the clean data.

Let r denote different proportions of noise. The value of r is from 0 to 1 with step 0.1. Let $\tilde{n}$ represent the number of instances of the UCI original datasets, then $\tilde{n} \times r$ represents the number of noisy that need to be generated. Uniformly distributed noise is randomly added into the original datasets to form new noisy datasets. Due to the limitation of pages, we only display the robustness comparison of these eight competing algorithms on datasets iris, seeds, and VC, as shown in Fig. 4.

When the noise ratios are relatively small, for example, $r < 0.2$, RSSC is slightly worse than other algorithms on ACC or NMI. However, with the increase of noise ratios ($r \geq 0.2$), RSSC performs better than others and gets the best NMI and ACC on these three datasets. No matter whether the trend is upward or downward, other comparison algorithms are consistently below the RSSC. Similar results can be obtained on the other seven datasets. Thus, RSSC can always get higher performance than these competing algorithms when the noise ratios are large enough ($r \geq 0.2$). These results demonstrate that RSSC is robust and can achieve excellent clustering performance on noisy datasets, even though noisy is relatively large.

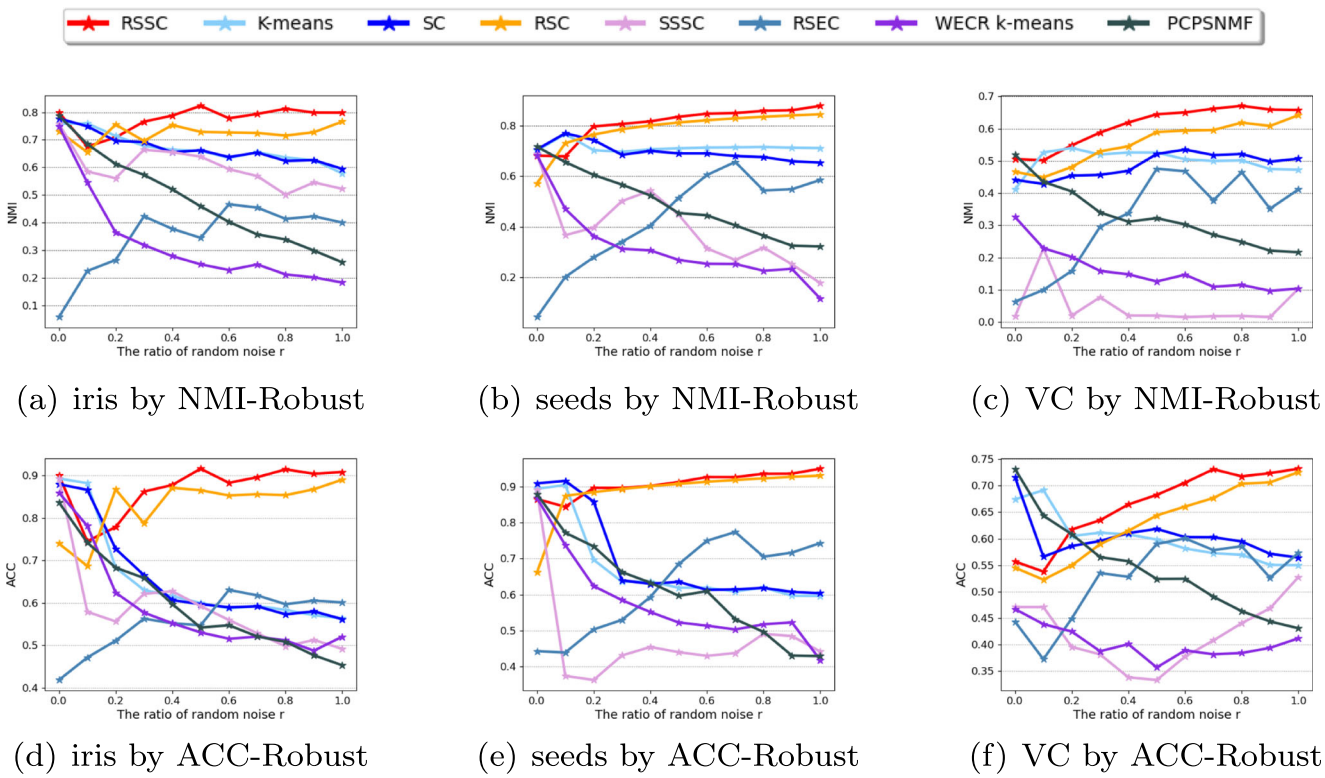
For RSSC, after mapping the original instances into new data space, the labeled data will be close to the canonical coordinate system, and the clean unlabeled data will be close to the labeled data with similar characteristics. Thus, the instances of the same cluster are closer to each other. RSSC uses p-nearest neighbors to construct a similarity



(a) iris by NMI (b) seeds by NMI (c) VC by NMI

(d) iris by ACC (e) seeds by ACC (f) VC by ACC

Fig. 3 The parameters analysis of p and μ on datasets iris, seeds, and VC



(a) iris by NMI-Robust (b) seeds by NMI-Robust (c) VC by NMI-Robust

(d) iris by ACC-Robust (e) seeds by ACC-Robust (f) VC by ACC-Robust

Fig. 4 Robustness comparison of eight competing algorithms on datasets iris, seeds and VC

matrix on the noise datasets. When the noise increase, the p neighbors of a specific noise instance are almost all noise points. Therefore, RSSC can perform well with the increase of noise. We choose $r = 0.4$ in the next experiments. In other words, in the following experiments, we add 40% noise into the clean datasets.

4.5 RSSC vs. its competing algorithms

In this section, we compare RSSC and its competing algorithms on sixteen datasets. Tables 2 and 3 show the NMI and ACC of these competing algorithms, while Table 4 shows their running time. Bold entries stand for the best-obtained score for their specific category.

To validate whether RSSC and its rivals have a significant difference in clustering performance, we conduct the Friedman test at a 95% significance level, under the null-hypothesis [8]. The performance of RSSC and that of its rivals has no significant difference if the null-hypothesis is accepted. If the null-hypothesis at the Friedman test is rejected, we proceed with the Nemenyi test as a post-hoc test. The p-value of Friedman test on NMI and ACC is 3.7971e-12 and 1.7098e-12, respectively. Therefore, there is a significant difference between these competing algorithms in cases of NMI and ACC. According to the Nemenyi test, the value of CD(critical difference) is 2.6249.

Table 5 shows the ranking of different algorithms in the evaluation results of NMI and ACC. A smaller value of ranks indicates a better performance of the algorithm. According to the rank-values of algorithms in Table 5, RSSC has the best performance in both cases of NMI and ACC. Meanwhile, Fig. 5 shows the Friedman test graphs of these eight competing algorithms in cases of NMI and ACC. Specifically, Fig. 5a shows no overlap between the horizontal line segments of RSSC and other competing algorithms, except RSC. Thus, RSSC is significantly better than WECR k-means, RSEC, SSSC, PCPSNMF, SC, k-means in the case of NMI. From Fig. 5b, RSSC is significantly better than other competing algorithms in the case of ACC, except RSC.

From Tables 2–5 and Fig. 5, we can draw:

- RSSC vs. WECR k-means: RSSC performs better than WECR k-means. On the average values of NMI and ACC, RSSC is 30.1% and 35.34% higher than WECR k-means, respectively. WECR k-means first samples multiple times from the original dataset and then uses k-means to generate cluster ensemble. Thus, WECR k-means must determine a cluster center for each cluster in advance, and these cluster centers determine the quality of the final clustering performance. When noise is considerable, the original label constraints will gradually change, and the clustering performance of WECR k-means will decrease. Instead, RSSC only uses label data to generate the matrix without constraints. Therefore, even if the noise increases, the method of forming the label matrix will not change, which leads to the robustness of RSSC.
- RSSC vs. RSEC: RSSC performs better than RSEC on all these datasets in both cases of NMI and ACC. On wine, seeds, and PDC, the NMI and ACC values of RSSC are at least 20% higher

Table 2 Clustering performance on sixteen datasets by NMI

Datasets	RSSC	WECR k-means	RSEC	SSSC	PCPSNMF	RSC	SC	k-means
iris	**0.7846 ± 0.01**	0.3783 ± 0.03	0.3961 ± 0.05	0.6306 ± 0.05	0.5205 ± 0.02	0.7752 ± 0.02	0.6901 ± 0.03	0.6278 ± 0.02
wine	**0.6822 ± 0.00**	0.5353 ± 0.02	0.4898 ± 0.07	0.5991 ± 0.00	0.5979 ± 0.03	0.6079 ± 0.09	0.6339 ± 0.00	0.5784 ± 0.03
plrx	**0.4074 ± 0.03**	0.0164 ± 0.00	0.0672 ± 0.05	0.0073 ± 0.00	0.0129 ± 0.01	0.0040 ± 0.00	0.0151 ± 0.01	0.0774 ± 0.00
seeds	**0.8228 ± 0.00**	0.3063 ± 0.02	0.4242 ± 0.04	0.5695 ± 0.03	0.5232 ± 0.00	0.8005 ± 0.00	0.6981 ± 0.01	0.7007 ± 0.01
VC	**0.6038 ± 0.01**	0.1477 ± 0.01	0.4561 ± 0.10	0.0180 ± 0.00	0.3117 ± 0.02	0.5281 ± 0.03	0.4291 ± 0.02	0.4165 ± 0.02
wdbc	**0.7411 ± 0.00**	0.4427 ± 0.01	0.4653 ± 0.09	0.4021 ± 0.00	0.4674 ± 0.01	0.7386 ± 0.00	0.7132 ± 0.00	0.6511 ± 0.01
PDC	**0.6751 ± 0.01**	0.3852 ± 0.00	0.4619 ± 0.04	0.4264 ± 0.00	0.4406 ± 0.00	0.5273 ± 0.11	0.0149 ± 0.00	0.3143 ± 0.01
banknote	**0.3713 ± 0.02**	0.0176 ± 0.00	0.2632 ± 0.08	0.0031 ± 0.00	0.1511 ± 0.00	0.2999 ± 0.08	0.1381 ± 0.01	0.1486 ± 0.01
yeast	**0.4431 ± 0.01**	0.3202 ± 0.00	0.4198 ± 0.01	0.3399 ± 0.01	0.3224 ± 0.04	0.4127 ± 0.01	0.3775 ± 0.01	0.3134 ± 0.02
Pt	**0.7328 ± 0.01**	0.4789 ± 0.00	0.7244 ± 0.02	0.5100 ± 0.00	0.5651 ± 0.02	0.6885 ± 0.01	0.5925 ± 0.00	0.5361 ± 0.01
USPS-01	**0.7709 ± 0.02**	0.3650 ± 0.01	0.6482 ± 0.04	0.6825 ± 0.02	0.5206 ± 0.02	0.6423 ± 0.00	0.5161 ± 0.01	0.5982 ± 0.00
MNIST-0	**0.5947 ± 0.05**	0.2819 ± 0.00	0.3305 ± 0.04	0.3031 ± 0.00	0.3111 ± 0.01	0.3828 ± 0.01	0.3607 ± 0.01	0.2903 ± 0.01
MNIST-012	0.6954 ± 0.00	0.4570 ± 0.01	**0.7099 ± 0.01**	0.4161 ± 0.00	0.4215 ± 0.03	0.6431 ± 0.01	0.4838 ± 0.03	0.5186 ± 0.03
MNIST-0123	**0.7243 ± 0.01**	0.4758 ± 0.00	0.7237 ± 0.01	0.4022 ± 0.19	0.5028 ± 0.02	0.6916 ± 0.00	0.4792 ± 0.02	0.4698 ± 0.00
Re0	**0.4383 ± 0.01**	0.3584 ± 0.03	0.2979 ± 0.02	0.3122 ± 0.02	0.2508 ± 0.00	0.2745 ± 0.11	0.2435 ± 0.01	0.2109 ± 0.02
Oh0	**0.5443 ± 0.02**	0.3561 ± 0.02	0.5361 ± 0.04	0.4983 ± 0.00	0.3076 ± 0.02	0.5277 ± 0.02	0.2002 ± 0.02	0.2773 ± 0.02
AVG.	**0.6374**	0.3327	0.4643	0.3825	0.3892	0.5340	0.4116	0.4206

Table 3 Clustering performance on sixteen datasets by ACC

Datasets	RSSC	WECR k-means	RSEC	SSSC	PCPSNMF	RSC	SC	k-means
iris	**0.8838 ± 0.01**	0.5524 ± 0.02	0.5552 ± 0.02	0.5848 ± 0.04	0.5976 ± 0.01	0.8810 ± 0.01	0.6190 ± 0.02	0.5405 ± 0.02
wine	**0.8024 ± 0.00**	0.5608 ± 0.03	0.4880 ± 0.06	0.5824 ± 0.00	0.5880 ± 0.02	0.7296 ± 0.07	0.7000 ± 0.00	0.6880 ± 0.02
plrx	**0.6431 ± 0.02**	0.3945 ± 0.01	0.5153 ± 0.06	0.4008 ± 0.01	0.4353 ± 0.00	0.3145 ± 0.02	0.3984 ± 0.03	0.3492 ± 0.00
seeds	**0.9088 ± 0.00**	0.5517 ± 0.02	0.5633 ± 0.04	0.5122 ± 0.01	0.6333 ± 0.00	0.9014 ± 0.00	0.6361 ± 0.01	0.6156 ± 0.01
VC	**0.6700 ± 0.04**	0.4009 ± 0.01	0.6553 ± 0.06	0.3406 ± 0.00	0.5572 ± 0.03	0.6005 ± 0.02	0.6083 ± 0.03	0.5729 ± 0.02
wdbc	**0.8758 ± 0.00**	0.4289 ± 0.01	0.7222 ± 0.05	0.4504 ± 0.00	0.5614 ± 0.01	0.8733 ± 0.00	0.8607 ± 0.00	0.8494 ± 0.00
PDC	**0.8272 ± 0.00**	0.4493 ± 0.02	0.5575 ± 0.04	0.5335 ± 0.00	0.5218 ± 0.04	0.5330 ± 0.04	0.5250 ± 0.00	0.5429 ± 0.00
banknote	**0.6974 ± 0.02**	0.3953 ± 0.01	0.5997 ± 0.06	0.3974 ± 0.00	0.5483 ± 0.03	0.5838 ± 0.03	0.5171 ± 0.01	0.5419 ± 0.01
yeast	**0.5372 ± 0.01**	0.2088 ± 0.00	0.3695 ± 0.02	0.3007 ± 0.00	0.3381 ± 0.05	0.5017 ± 0.01	0.2930 ± 0.01	0.2964 ± 0.02
Pt	**0.7989 ± 0.01**	0.1688 ± 0.01	0.7428 ± 0.05	0.2881 ± 0.00	0.4119 ± 0.02	0.6963 ± 0.02	0.5755 ± 0.01	0.4323 ± 0.02
USPS-01	**0.9019 ± 0.02**	0.6740 ± 0.02	0.7503 ± 0.10	0.8655 ± 0.01	0.6613 ± 0.01	0.6371 ± 0.00	0.6668 ± 0.02	0.7919 ± 0.00
MNIST-0	**0.9320 ± 0.02**	0.6598 ± 0.01	0.5884 ± 0.05	0.7694 ± 0.00	0.7068 ± 0.02	0.8342 ± 0.01	0.7916 ± 0.01	0.7350 ± 0.00
MNIST-012	**0.7537 ± 0.02**	0.6759 ± 0.01	0.7335 ± 0.00	0.2936 ± 0.00	0.5678 ± 0.04	0.7292 ± 0.01	0.6587 ± 0.01	0.6402 ± 0.01
MNIST-0123	**0.7676 ± 0.01**	0.6900 ± 0.00	0.7506 ± 0.01	0.2183 ± 0.00	0.6209 ± 0.03	0.7378 ± 0.00	0.6423 ± 0.02	0.6687 ± 0.03
Re0	0.5083 ± 0.01	**0.5265 ± 0.02**	0.3700 ± 0.12	0.3671 ± 0.01	0.2673 ± 0.02	0.4181 ± 0.01	0.2673 ± 0.03	0.4711 ± 0.01
Oh0	**0.5247 ± 0.02**	0.3874 ± 0.03	0.5114 ± 0.02	0.3820 ± 0.02	0.2239 ± 0.30	0.4397 ± 0.00	0.2239 ± 0.02	0.4048 ± 0.03
AVG.	**0.7520**	0.4828	0.5920	0.4554	0.5150	0.6527	0.5488	0.5713

than that of RSEC. RSEC jointly learns a robust representation for the co-association matrix through low-rank constraint and finds the final partition in a unified optimization framework. However, RSEC needs to solve n optimization sub-problems on n data points. The eigenvector is updated through each iteration by the Singular Value Thresholding (SVT) operator, and the clustering results are affected by the increase in the number of iterations. By contrast, RSSC does not require iterative operations and only runs once. It is evident from Table 4 that RSSC is much faster than RSEC on running time.

– RSSC vs. SSSC: RSSC gets much higher performance than SSSC on these datasets in both cases of NMI and ACC. SSSC is a semi-supervised spectral clustering method consistent with labeled and unlabeled data

Table 4 Run Time(sec.) of all the compared methods on sixteen datasets

Datasets	RSSC	WECR k-means	RSEC	SSSC	PCPSNMF	RSC	SC	k-means
iris	0.0845	0.2423	13.0595	0.9545	0.9005	0.0611	0.0278	0.2621
wine	0.0975	0.2810	14.1015	1.0508	0.9711	0.0658	0.0253	0.0206
plrx	0.1259	0.1948	15.4322	0.9202	1.2904	0.0555	0.0315	0.0222
seeds	0.1461	0.2609	14.6580	1.0806	1.6806	0.0722	0.0347	0.0221
VC	0.3285	0.2080	17.1694	1.3292	7.7209	0.0694	0.0426	0.0266
wdbc	1.1120	0.3558	24.5812	1.9540	24.8306	0.7490	0.0824	0.0303
PDC	0.3285	1.4644	17.1694	6.7739	50.4626	0.0694	0.0426	0.0854
banknote	10.1039	2.3363	66.7948	6.1065	244.8491	0.7419	0.3840	0.0523
yeast	14.8369	4.3403	83.7529	8.3654	265.2536	0.4008	0.6299	0.1501
Pt	180.6484	40.4137	514.9024	30.7948	1549.6428	1.1341	2.6991	0.1369
USPS-01	3.8348	4.3563	64.6003	12.9925	70.5237	0.6907	0.1770	0.0521
MNIST-0	4.7833	2.8742	138.6816	14.5841	63.8813	1.5594	0.1823	0.0967
MNIST-012	10.8649	5.5957	270.9705	0.0001	148.2582	2.6832	0.3666	0.1636
MNIST-0123	30.0945	8.8905	397.9122	71.1581	285.1119	6.7335	0.9323	0.3181
Re0	8.3742	3.7885	623.9524	217.3313	718.6201	5.4825	0.3426	0.6812
Oh0	3.1334	2.4154	380.0712	75.5827	252.5800	2.3633	0.1920	0.5388
AVG.	16.9285	5.4532	169.5826	45.6217	230.4111	1.5135	0.3970	**0.1662**

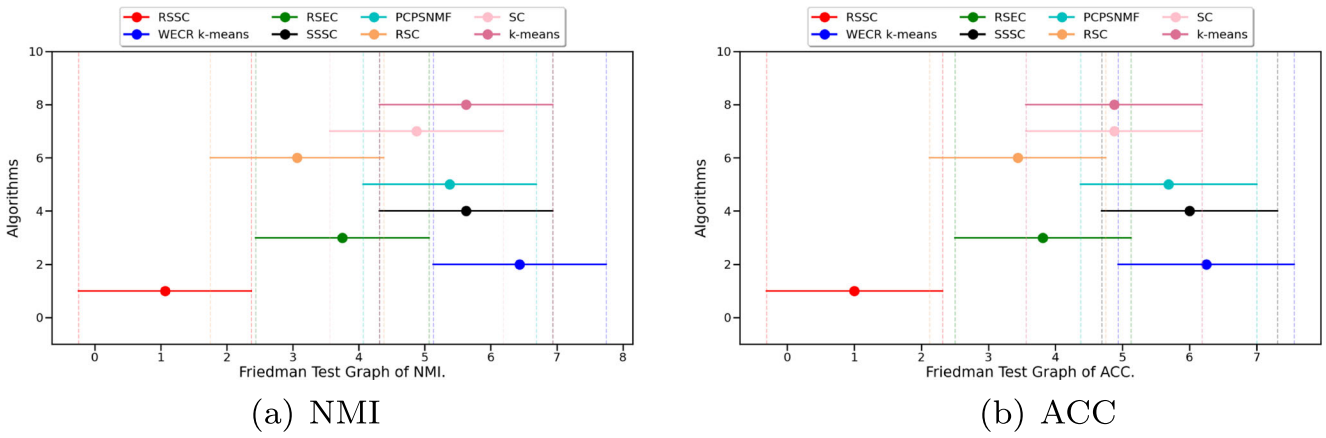
Table 5 The ranks of these competing algorithms by NMI and ACC

Datasets	RSSC		RSEC		WECR k-means		SSSC		PCPSNMF		RSC		SC		k-means	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
iris	1	1	7	6	8	7	4	5	6	4	2	2	3	3	5	8
wine	1	1	8	8	7	7	4	6	5	5	3	2	2	3	6	4
plrx	1	1	3	2	4	6	7	4	6	3	8	8	5	5	2	7
seeds	1	1	7	6	8	7	5	8	6	4	2	2	4	3	3	5
VC	1	1	3	2	7	7	8	8	6	6	2	4	4	3	5	5
wdbc	1	1	5	5	6	8	7	7	5	6	2	2	4	3	3	4
PDC	1	1	3	2	6	8	5	5	4	7	2	4	7	6	8	3
banknote	1	1	3	2	7	8	8	7	4	4	2	3	6	6	5	5
yeast	1	1	2	3	7	8	5	5	6	4	3	2	4	7	8	6
Pt	1	1	2	2	8	8	7	7	5	6	3	3	4	4	6	5
USPS-01	1	1	3	4	8	5	2	2	6	7	4	8	7	6	5	3
MNIST-0	1	1	5	8	8	7	6	4	4	6	2	2	3	3	7	5
MNIST-012	2	1	1	2	6	4	8	8	7	7	3	3	5	5	4	6
MNIST-0123	1	1	2	2	6	4	7	8	4	7	3	3	5	6	8	5
Re0	1	2	4	5	2	1	3	6	6	7	5	4	7	8	8	3
Oh0	1	1	2	2	5	5	4	6	6	8	3	3	8	7	7	4
AVG. RANK	1.0625	1.0625	3.7500	3.8125	6.4375	6.2500	5.6250	6.0000	5.3750	5.6875	3.0625	3.4375	4.8750	4.8750	5.6250	4.8750

that can perform well on clean datasets. However, SSSC does not consider noise, and it performs worse on these noise datasets. Thus, removing noise in the clustering process is necessary because these noise data can significantly influence performance. Besides, SSSC uses a fully connected undirected weighted graph when constructing the similarity matrix. Thus, SSSC spends more time than RSSC, which uses the p-nearest neighbor graph to construct the similarity matrix.

– RSSC vs. PCPSNMF: RSSC performs better than PCPSNMF in both cases of NMI and ACC. PCPSNMF is a symmetric NMF-based semi-supervised clustering method that took advantage of the supervisory information by using them to construct a superior affinity matrix. However, PCPSNMF is not robust to noise. On noise datasets, the similarity matrice structure of PCPSNMF will be destroyed by noise, thereby affecting the assignment matrix's generation. In addition, the iterative calculation of PCPSNMF makes the running time much longer than RSSC. Therefore, anti-noise processing for semi-supervised clustering is necessary.

– RSSC vs. RSC: RSSC outperforms RSC on ten of sixteen datasets in both cases of NMI and ACC. On plrx and PDC, the NMI and ACC values of RSSC are at least 26% higher than RSC. RSC uses a sparse and latent decomposition of the similarity graph during



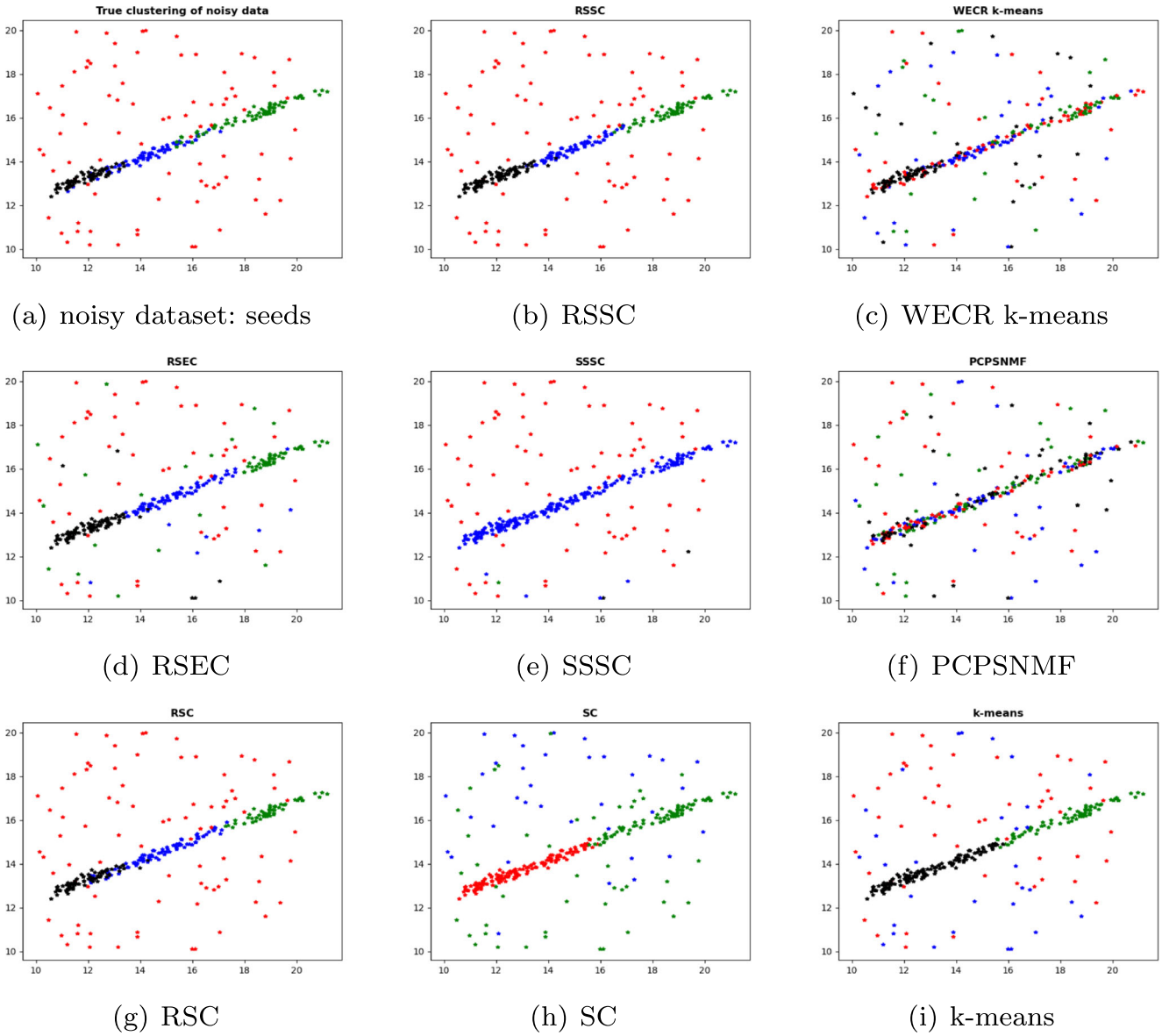
(a) NMI (b) ACC

Fig. 5 Statistical test graphs of these eight competing algorithms in cases of NMI and ACC

spectral clustering. Meanwhile, RSC requires each local node to have a minimum degree in each iteration and calculates truncated eigendecomposition that affects the final clustering performances. In contrast, RSSC acts on the entire dataset and has only one global minimum solution. On running time, RSC is faster than RSSC. RSSC has a matrix inversion process, while RSC uses the potential decomposition of the similarity graph to separate the clean dataset from the noise.

– RSSC vs. SC, k-means: Compared with the basic clustering algorithm, RSSC increased 22.58% and 20.32% than SC on the average of NMI and ACC, respectively. Meanwhile, RSSC increased 21.68% and 18.03% respectively than k-means. SC and k-means are sensitive to noise, leading to a significant decrease in clustering performance. SC and k-means are much faster on running time than RSSC due to their simplicity.

In summary, RSSC performs much better than its competing algorithms on these noise datasets in both cases of NMI and ACC. The experimental results thoroughly verify our new algorithm's effectiveness and the necessary of anti-noise processing for semi-supervised clustering.



(a) noisy dataset: seeds (b) RSSC (c) WECR k-means

(d) RSEC (e) SSSC (f) PCPSNMF

(g) RSC (h) SC (i) k-means

Fig. 6 The clustering results of RSSC and other seven algorithms on the noisy dataset seeds

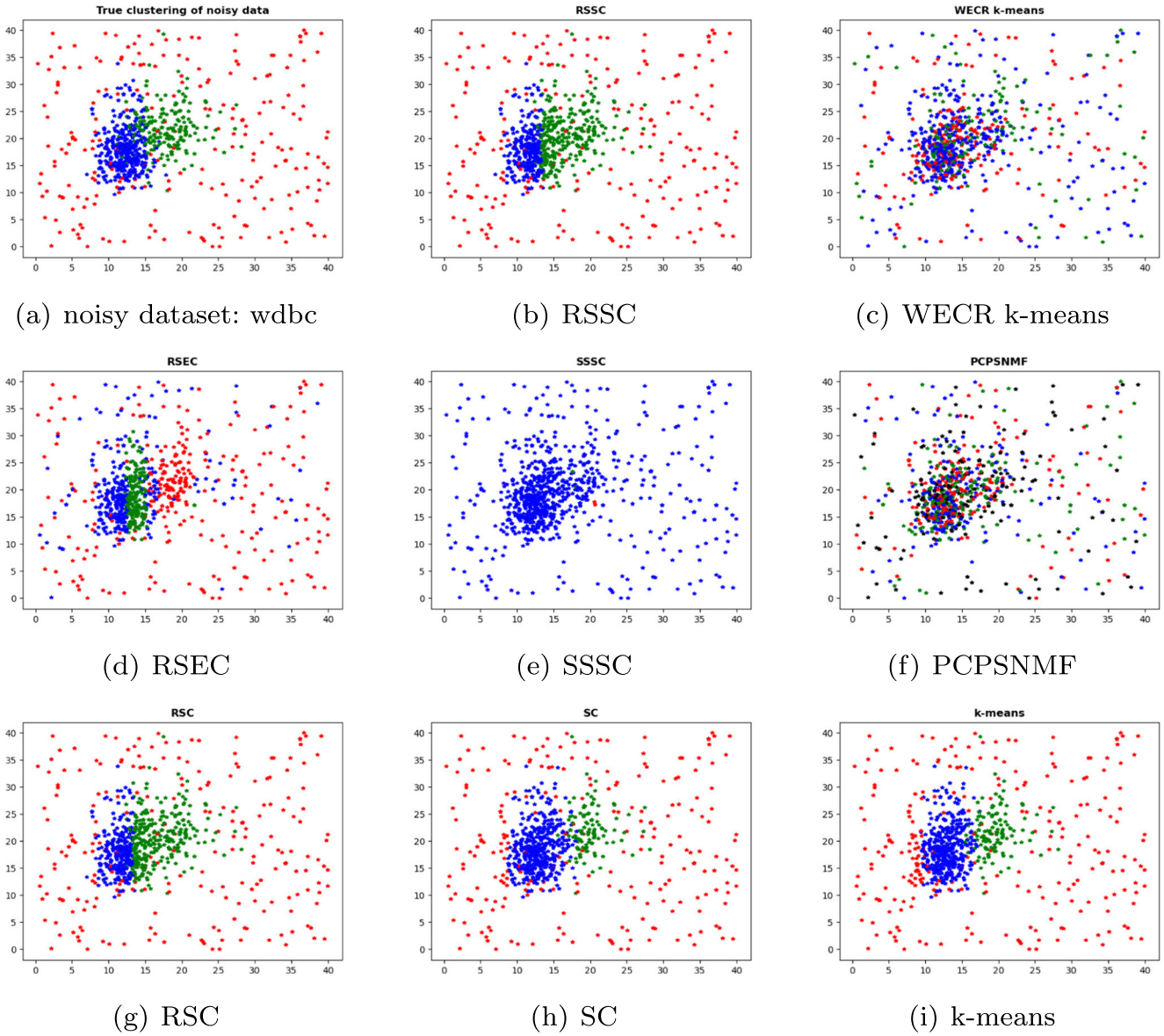
4.6 Graphical display

In this part, to visually present the results of these competing algorithms, we offer the effect diagrams of these eight algorithms on two datasets: seeds and wdbc. The clustering results are presented in a two-dimensional form, as shown in Figs. 6a and 7a, where the red dots represent the 40% added noise, and the other colors (black, green, and blue) represent the correct clustering results of the clean data.

RSSC and the other seven competing algorithms are clustered on the same two noise datasets, where Figs. 6a and 7a are the correct clustering results. It can be seen intuitively from Figs. 6 and 7 that RSSC is the closest to the correct clustering results. These figures intuitively demonstrate the effectiveness and robustness of the RSSC algorithm.

5 Conclusions

This paper proposes a novel algorithm (RSSC) to achieve robust clustering on semi-supervised datasets. In terms of transductive warping, RSSC can make full use of label information to guide those unlabeled data in the same



(a) noisy dataset: wdbc (b) RSSC (c) WECR k-means

(d) RSEC (e) SSSC (f) PCPSNMF

(g) RSC (h) SC (i) k-means

Fig. 7 The clustering results of RSSC and other seven algorithms on the noisy dataset wdbc

cluster to close it. Meanwhile, for the noisy data that has no label guidance, they will form noise clusters. RSSC can simultaneously identify noise and complete data clustering tasks even for high noise ratio datasets. Meanwhile, RSSC constructs a convex objective function that can obtain the optimal global solution. Extensive experiments verify the effectiveness and robustness of RSSC. The limitation of RSSC is that the number of clusters needs to be manually defined, and sometimes it may be more troublesome if we do not know how many clusters in advance. Besides, with the rapid increase in data volume, algorithms in batch mode can no longer meet time and space complexity requirements. Thus, our future work will focus on cluster analysis for online streaming data.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China under grants 61906056, 61876001.

References

1. Aggarwal CC (2018) An introduction to cluster analysis. In: Data clustering, Chapman and hall/CRC, pp 1–28
2. Alok AK, Saha S, Ekbal A (2017) Semi-supervised clustering for gene-expression data in multiobjective optimization framework. Int J Mach Learn Cybern 8(2):421–439
3. Amigó E, Gonzalo J, Artiles J, Verdejo F (2009) A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4):461–486
4. Ammour A, Aouraghe I, Khaissidi G, Mrabti M, Aboulem G, Belahsen F (2020) A new semi-supervised approach for characterizing the arabic on-line handwriting of parkinson's disease patients. Comput Methods Prog Biomed 183:104979
5. Baghshah MS, Shouraki SB (2009) Semi-supervised metric learning using pairwise constraints. In: Twenty-first international joint conference on artificial intelligence, pp 1217–1222
6. Eric B (2013) Semi-supervised clustering methods. Wiley Interdisciplinary Reviews: Computational Statistics 5(5):349–361
7. Bojchevski A, Matkovic Y, Günnemann S (2017) Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 737–746
8. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30
9. Fang X, Yong Xu, Li X, Lai Z, Wong WK (2015) Robust semi-supervised subspace clustering via non-negative low-rank representation. IEEE Trans Cybern 46(8):1828–1838
10. Hariri S, Kind MC, Brunner RJ (2021) Extended isolation forest. IEEE Trans Knowl Data Eng 33(4):1479–1489
11. Ienco D, Pensa RG (2018) Semi-supervised clustering with multiresolution autoencoders. In: 2018 International joint conference on neural networks (IJCNN), IEEE, pp 1–8
12. Ionescu C, Popa A, Sminchisescu C (2017) Large-scale data-dependent kernel approximation. In: Artificial intelligence and statistics, PMLR, pp 19–27
13. Kang Z, Shi G, Huang S, Chen W, Pu X, Zhou JT, Xu Z (2020) Multi-graph fusion for multi-view spectral clustering. Knowl-Based Syst 189:105102
14. Kim Y, Do H, Kim SB (2020) Outer-points shaver: Robust graph-based clustering via node cutting. Pattern Recogn 97:107001
15. Kong W, Hu S, Zhang J, Dai G (2013) Robust and smart spectral clustering from normalized cut. Neural Comput Applic 23(5):1503–1512
16. Lai Y, He S, Lin Z, Yang F, Zhou Qi-Feng, Zhou X (2019) An adaptive robust semi-supervised clustering framework using weighted consensus of random k-means ensemble. IEEE Trans Knowl Data Eng 33(5):1877–1890
17. Li X, Yin H, Ke Z, Zhou X (2020) Semi-supervised clustering with deep metric learning and graph embedding. World Wide Web 23(2):781–798
18. Li Z, Liu J, Chen S, Tang X (2007) Noise robust spectral clustering. In: 2007 IEEE 11Th international conference on computer vision, IEEE Computer Society, pp 1–8
19. Liu H, Li J, Yue Wu, Yun Fu (2019) Clustering with outlier removal. IEEE Transactions on Knowledge and Data Engineering 33(6):2369–2379
20. Ma Y, Ganapathiraman V, Zhang X (2019) Learning invariant representations with kernel warping. In: The 22nd international conference on artificial intelligence and statistics, PMLR, pp 1003–1012
21. Mai X, Couillet R (2018) Semi-supervised spectral clustering. In: 2018 52Nd asilomar conference on signals, systems, and computers, IEEE, pp 2012–2016
22. Maulik U, Bandyopadhyay S (2002) Performance evaluation of some clustering algorithms and validity indices. IEEE Trans Pattern Anal Mach Intell 24(12):1650–1654
23. Ning J, Chen L, Chen J (2018) Relative density-based outlier detection algorithm. In: Proceedings of the 2018 2nd international conference on computer science and artificial intelligence, ACM, pp 227–231
24. Ott L, Pang L, Ramos FT, Chawla S (2014) On integrated clustering and outlier detection. Advances in Neural Information Processing Systems 27:1359–1367
25. Peng S, Ser W, Chen B, Lin Z (2021) Robust semi-supervised nonnegative matrix factorization for image clustering. Pattern Recogn 111:107683
26. Qian H, Pan SJ, Miao C (2019) Distribution-based semi-supervised learning for activity recognition. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 7699–7706
27. Rossi RG, Marcacini RM, Rezende SO et al (2013) Benchmarking text collections for classification and clustering tasks. Tech. rep. 395, Institute of Mathematics and Computer Sciences University of Sao Paulo
28. Sanodiya RK, Saha S, Mathew J (2019) A kernel semi-supervised distance metric learning with relative distance: Integration with a moo approach. Expert Syst Appl 125:233–248
29. Shen P, Du X, Li C (2016) Distributed semi-supervised metric learning. IEEE Access 4:8558–8571
30. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
31. Śmieja M, Struski Ł, Figueiredo MAT (2020) A classification-based approach to semi-supervised clustering with pairwise constraints. Neural Netw 127:193–203
32. Smola AJ, Kondor R (2003) Kernels and regularization on graphs. In: Learning theory and kernel machines, vol 2777, Springer, pp 144–158
33. Bo T, He H (2017) A local density-based approach for outlier detection. Neurocomputing 241:171–180
34. Tang Y, Wang J, Gao B, Dellandréa E, Gaizauskas R, Chen L (2016) Large scale semi-supervised object detection using visual and semantic knowledge transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2119–2128

35. Tao Z, Liu H, Li S, Ding Z, Yun Fu (2019) Robust spectral ensemble clustering via rank minimization. ACM Transactions on Knowledge Discovery from Data (TKDD) 13(1):4:1–4:25
36. Engelen JEV, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440
37. Vladimir V (2006) Transductive inference and semi-supervised learning. In: Semi-supervised learning, The MIT Press, pp 452–472
38. Veras R, Aires K, Britto L et al (2018) Medical image segmentation using seeded fuzzy c-means: a semi-supervised clustering algorithm. In: 2018 International joint conference on neural networks (IJCNN), IEEE, pp 1–7
39. Wang F, Sun J (2015) Survey on distance metric learning and dimensionality reduction in data mining. Data Mining and Knowledge Discovery 29(2):534–564
40. Wang J, Tian F, Liu CH, Wang X (2015) Robust semi-supervised nonnegative matrix factorization. In: 2015 International joint conference on neural networks (IJCNN), IEEE, pp 1–8
41. Wu W, Jia Y, Kwong S, Hou J (2018) Pairwise constraint propagation-induced symmetric nonnegative matrix factorization. IEEE Transactions on Neural Networks and Learning Systems 29(12):6348–6361
42. Xu X, Ding S, Wang L, Wang Y (2020) A robust density peaks clustering algorithm with density-sensitive similarity. Knowl-Based Syst 200:106028
43. Xu Z, Ke Y (2016) Effective and efficient spectral clustering on text and link data. In: Proceedings of the 25th ACM international on conference on information and knowledge management, ACM, pp 357–366
44. Lu Y, Liu Y (2018) Ensemble biclustering gene expression data based on the spectral clustering. Neural Comput Applic 30(8):2403–2416
45. Yu Z, Luo P, Liu J, Wong H-S, You J, Han G, Zhang J (2018) Semi-supervised ensemble clustering based on selected constraint projection. IEEE Trans Knowl Data Eng 30(12):2394–2407
46. Zhou Z, Si G, Zhang Y, Zheng K (2018) Robust clustering by identifying the veins of clusters based on kernel density estimation. Knowl-Based Syst 159:309–320
47. Zhu X (2017) Semi-supervised learning. In: Encyclopedia of machine learning and data mining, Springer, pp 1142–1147
48. Zhu X, Kandola JS, Ghahramani Z, Lafferty JD (2004) Nonparametric transforms of graph kernels for semi-supervised learning. In: Advances in neural information processing systems, vol 17, pp 1641–1648

Peng Zhou received the Ph.D. degree from the Hefei University of Technology, Hefei, China, in 2018. He is currently a Lecturer with Anhui University, Hefei. His research interests include machine learning, data mining and knowledge engineering.

Ni Wang received the B.Sc. degree from Wannan Medical College, WuHu, China, in 2015. She is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, Anhui University, Hefei. Her current research interests include stream data mining and robust clustering.

Shu Zhao received her Ph.D. degree in Computer Science from Anhui University in 2007. She is currently a professor in the Department of Computer Science and Technology, Anhui University. Her current research interests include quotient space theory, granular computing, data mining and machine learning.

Yanping Zhang received her PhD degree in Computer Science from Anhui University in 2003. She is currently a professor in the Department of Computer Science and Technology, Anhui University. Her current research interests include deep learning, quotient space theory, granular computing, and machine learning.