Robust large-scale clustering based on correntropy

Guodong Jin, Jing Gao*, Lining Tan*
Rocket Force Engineering University, Xian, Shannxi, China
* These authors contributed equally to this work.
* tanlining021@hotmail.com

Abstract

With the explosive growth of data, how to efficiently cluster large-scale unlabeled data has become an important issue that needs to be solved urgently. Especially in the face of large-scale real-world data, which contains a large number of complex distributions of noises and outliers, the research on robust large-scale real-world data clustering algorithms has become one of the hottest topics. In response to this issue, a robust large-scale clustering algorithm based on correntropy (RLSCC) is proposed in this paper, specifically, k-means is firstly applied to generated pseudo-labels which reduce input data scale of subsequent spectral clustering, then anchor graphs instead of full sample graphs are introduced into spectral clustering to obtain final clustering results based on pseudo-labels which further improve the efficiency. Therefore, RLSCC inherits the advantages of the effectiveness of k-means and spectral clustering while greatly reducing the computational complexity. Furthermore, correntropy is developed to suppress the influence of noises and outlier the real-world data on the robustness of clustering. Finally, extensive experiments were carried out on real-world datasets and noise datasets and the results show that compared with other state-of-the-art algorithms, RLSCC can improve efficiency and robustness greatly while maintaining comparable or even higher clustering effectiveness.

Introduction

As the core of artificial intelligence and data science, machine learning is a discipline that aims at developing learning algorithms that build models from data (experience). In the past decades, machine learning has made great progress, and abundant techniques based on it have emerged. These techniques have played an important role in various practical applications, such as image processing [1–5], environmental monitoring [6–14], and data mining [15–25]. Among these techniques, clustering is currently one of the most popular topics in machine learning, which can automatically divide unlabeled data into different groups (clusters). In the past decades, scholars have proposed lots of impressive works. However, with the advent of the information age, clustering is bothered by several challenges. On the one hand, with the exponential rise of data, conventional clustering algorithms are finding it challenging to deal with these massive amounts of unlabeled data. The issue of how to efficiently cluster these massive
amounts of unlabeled data has emerged as a critical challenge in unsupervised learning. On the other hand, in real-world clustering activities, most data contain various complex noises and outliers, which have a substantial negative impact on clustering robustness. Hence another significant problem that we should be concerned with is how to enhance the robustness of clustering algorithms in the face of real-world data. Based on the above-mentioned challenges and problems, researchers have made a lot of efforts to find a way out.

To improve the efficiency of clustering large-scale data, many accelerated clustering algorithms have been proposed with different strategies. They can be divided into k-means-based methods [15–19] and anchor graph-based methods [20–25]. As the most common acceleration algorithm, the k-means-based algorithm, which is proved to be equivalent to the algorithm based on matrix factorization, has linear computational complexity and better clustering performance. For example, FNMTF [16] and LP-FNMTF [16] proposed by Wang et al. directly constrain the factor matrix as the cluster indicator matrix to avoid additional operations when the optimization is completed. Furthermore, on this basis, Han et al. proposed a more efficient algorithm named BKM [17] to further constrain the absorption factor to a diagonal matrix to reduce the computational complexity. These k-means-based algorithms meet the efficiency requirements for processing large-scale data to a certain extent, but their direct processing of the original data makes their efficiency very sensitive to the data dimension. When the data dimension is high, their efficiency will decrease significantly [26]. As for the anchor graph-based methods, they are inspired by the idea of spectral clustering and construct anchor graphs instead of traditional full sample graphs to reduce computational complexity. Compared with traditional spectral clustering, anchor graph-based methods can greatly improve clustering efficiency while maintaining comparable clustering effectiveness, but they are still time-consuming due to the large amount of time needed to process the obtained anchor graphs. For example, ULGE [20] uses an effective method to construct a similarity matrix and then efficiently performs spectral analysis. FSCAG [21] constructs an anchor graph that takes into account spectral and spatial characteristics and performs spectral analysis to process large-scale hyperspectral images. SCHBG [22] explores the pyramid structure by a novel type of spectral clustering based on hierarchical bipartite graphs is proposed. Most of the above-mentioned anchor graph-based algorithms improve efficiency by optimizing the constructing anchor graph part, but they still have high complexity when performing spectral analysis on the obtained graphs, so it is difficult to directly apply them to those large-scale clustering tasks with higher efficiency requirements. Based on this, it is still urgently needed to develop an efficient large-scale clustering algorithm that is insensitive to data dimensions.

As for improving the robustness of real-world data clustering tasks, it is currently widely adopted to use the robustness norm to measure the error between the original data and the reconstructed representation. For example, the $L_1$-norm-based methods [27, 28] and the method based on the $L_{21}$-norm-based methods [29–31]. LSSC [27] uses the $L_1$-norm to define a sparse coding problem to improve the robustness of the representation, RDCF [28] uses the $L_1$-norm to minimize the error before and after the conceptual decomposition, and the $L_{21}$-norm is used to select features to constrain row sparsity by enhancing the matrix and constrain the errors of the subspace representation and the original data in LSS [29] and LRR [30], respectively. Although these algorithms based on $L_1$-norm and $L_{21}$-norm can suppress simple noise better, their robustness will be significantly reduced when the noise distribution is more complex. Recently, correntropy [32], a robust local measurement criterion in information theory learning (ITL), has been introduced into clustering and has achieved good robustness [33–38], such as GCCF [34], CHNMF [35] and CSNMF [36]. However, they cannot be applied to large-scale clustering tasks due to their square or even cubic complexity. Therefore, how to
introduce correntropy into large-scale real-world data clustering task to improve clustering robustness has become an important task at present.

To address the above problems, we propose a robust large-scale clustering algorithm based on correntropy (RLSCC). In the RLSCC model, for improving efficiency, pseudo-labels generated by k-means rather than the original data are utilized as the input of subsequent spectral clustering which greatly reduces the data scale involved in subsequent operations. Then anchor graph clustering instead of traditional spectral clustering is performed based on the obtained pseudo-labels to directly get the sample category which further accelerates the model. In terms of robustness, correntropy is applied in the model to suppress the impact of complex noises and outliers. The main contributions of this paper are summarized as follows:

• A novel robust large-scale clustering algorithm based on correntropy (RLSCC) is proposed in this paper. Compared with most accelerated methods which are mainly k-means-based methods and anchor graph-based methods, RLSCC is much more insensitive to data dimensions than k-means-based methods due to the implementation of pseudo-labels and graph learning while saving more time of subsequent spectral analysis than anchor graph-based methods by directly getting the sample class. Furthermore, correntropy is applied in our model to improve robustness.

• A novel optimization strategy based on half-quadratic (HQ) minimization technique [39–41] is proposed in this paper to solve the non-convex objective function of RLSCC owing to the introduction of correntropy, which can improve the efficiency as well by a few number of iterations. In addition, the complexity and parameter sensitivity of RLSCC are also analyzed.

• Extensive experiments have been performed on different real-world datasets and the results show that compared with the current mainstream fast clustering, RLSCC can efficiently obtain better performance than these algorithms.

The remainder of this paper is organized as follows: A novel robust large-scale clustering method named RLSCC is proposed in Section II. An iterative strategy is proposed for solving RLSCC and its computational complexity analysis is shown in Section III. Then, Section IV shows some experiment details and Section V is the conclusion.

Methodology
To improve the clustering efficiency and robustness of large-scale real-world data clustering tasks, we propose a robust large-scale clustering method based on correntropy (RLSCC). This section will give a detailed description of the process of the RLSCC model.

Pseudo-labels generation
Consider a data matrix $X \in \mathbb{R}^{N \times D}$, where $N$ is the number of samples and $D$ is the number of dimensions. Put it into k-means model as follows:

$$
\min_{W \in \text{IND} \cdot C} \|X - WC\|^2_F
$$

(1)

where $W \in \mathbb{R}^{N \times C}$ is a indicator matrix where $W_{i,j} = 1$ if $i$th sample is clustered into category $j$th otherwise $W_{i,j} = 0$, $C \in \mathbb{R}^{C \times D}$ is the cluster center matrix, whose each row represent a cluster center.

After we get $W$ based on $X$ from k-means, $W$ is regarded as pseudo-labels to participate in the follow-up process. This step successfully compress the original data with $N \times D$ scale into a
Anchor graph learning

Spectral clustering can often obtain better clustering effectiveness because it is not limited by the sample space shape and the use of sample spatial geometric information, but traditional spectral clustering is often difficult to be applied in large-scale clustering tasks due to its high computational complexity which usually is quadratic or cubic [42]. Based on this problem, anchor strategy has been developed and has been widely used in many graph learning works. This subsection gives some details about the process of anchor graph learning.

**Anchors generation.** There are currently two main methods for generating anchors which are random sampling and k-means, respectively. Because k-means can often provide better clustering performance under the same number of anchors than random sampling, this work uses k-means to coarsely cluster the original data to get representative anchors for the graph constructing part of spectral clustering.

**Anchor graph construction.** After getting all anchors defined as \( s_1, \ldots, s_M \) in our work, the anchor graph needs to be constructed between the samples set and the anchors set. Traditional anchor graph construction methods usually include: 1) Calculate the distance between all points in the samples set and all points in the anchors set to directly obtain a distance matrix; 2) Set a fixed threshold, let the distance less than it be 1, and the rest are all 0; 3) Set a fixed threshold, the distance less than it remains value, and the rest is set to 0. Although these methods can obtain anchor graph and have applied in many cases, their exploration of the geometric structure of the sample space is limited. In our work, following [43, 44], a normalized KNN anchor graph is constructed by using the first k-nearest neighbors of a fixed sample as follows:

\[
A^{(i)} = \frac{\hat{D}(x^{(i)}, s^{(i)}) - \hat{D}(x^{(i)}, s^{(i+1)})}{\sum_{j=1}^{k} (\hat{D}(x^{(i)}, s^{(i+1)}) - \hat{D}(x^{(i)}, s^{(i)}))} \quad A^{(j)} = \frac{\hat{D}(x, s_{j+1}) - \hat{D}(x, s_{j})}{\sum_{j=1}^{k} (\hat{D}(x, s_{j+1}) - \hat{D}(x, s_{j}))}
\]

where \( \hat{D}(x, s) \) is a sort function that can sort the distance in ascending order, and it satisfies \( \hat{D}(x, e) = \phi(D(x, e)) \), and \( D(x, e) = \|x - e\|^2 \). The property of \( \sum_{j=1}^{k} s_{j} = 1 \) can get a more meaningful clustering performance.

Anchor graph based clustering with pseudo-labels

To inherit the high efficiency of k-means and the high effectiveness of graph-based clustering, a fast clustering model is proposed in this subsection, which uses the correntropy to minimize the clustering results of k-means and graph-based clustering to ensure the clustering effectiveness and robustness, while greatly improving the efficiency of clustering. The objective function of RLSCC can be defined as the following form:

\[
\max_{U_p} Q(U_p) = \sum_{i=1}^{N} G(\sqrt{\sum_{i=1}^{K} (W_{q} - (Z_{p} U_{p})_{i})^2})
\]

where \( G(\cdot) \) represents the kernel function of correntropy, and \( Z_{p} \) is the first \( p \) columns of \( Z \).
which denotes the best represent the structure of the sample graph. $Z$ can be defined as:

$$[Z, V, H] = \text{svd}(L)$$

(4)

where $L$ is the Laplacian matrix of anchor graph $A$ and it can be defined as:

$$L = I - \frac{1}{2}SD^{-1/2}S$$

(5)

where $S = A^T A$ denotes the similarity matrix of $A$, and $D$ is the degree matrix of $A$ satisfies $d_{ii} = \sum_j s_{ij}$.

**Optimization and analysis**

**Optimization**

In this subsection, an iterative optimization method is proposed to solve the objective function. Note that Eq (3) is a non-convex function. To bring the optimization problem into a convex situation, we use half-quadratic technology to transform Eq (3) into the following formula:

$$\min_{U_p, V} O = \sum_{i=1}^{N} \left( v_i \left( \sum_{j=1}^{K} (W_{ij} - (Z_p U_p)_{ij})^2 \right) \right)$$

(6)

where $V$ is a diagonal matrix, and its $i$th diagonal element $V_{ii} = v_i$ can be given as:

$$v_i = \frac{G(\sum_{j=1}^{K} (W_{ij} - (Z_p U_p)_{ij})^2)}{\sigma^2}$$

$$= \frac{\exp(-\sum_{j=1}^{K} (W_{ij} - (Z_p U_p)_{ij})^2/2\sigma^2)}{\sigma^2}$$

(7)

where $\sigma$ is a free parameter that controls the robustness of the correntropy.

Now, the objective function Eq (6) can be solved directly, and the proposed optimization formulation contains two variants totally. Here, we fix one of them and update the other one. In practice, the iterative optimization performs two steps: $V$-step and $U_p$-step. The specific steps are as follows:

**V-step.** Fixing $U_p$, $V$ can be updated as the following formula:

$$V_{ii} = v_i = \exp \left( \frac{-\sum_{j=1}^{K} (W_{ij} - (Z_p U_p)_{ij})^2}{2\sigma^2} \right) / \sigma^2$$

(8)

**U_p-step.** When $V$ is fixed, Eq (6) can be transformed into:

$$\min_{U_p} O = Tr((W - Z_p U_p)^T V(W - Z_p U_p))$$

$$= Tr(W^T V W) - 2Tr(U_p^T Z_p^T WV) + Tr(U_p^T Z_p^T VZ_p U_p)$$

(9)
Assuming that $\Phi = [\Phi_{ij}] \in \mathbb{R}^{N \times N}$ are Lagrange multipliers, the Lagrange function of Eq (6) can be expressed by the following formula:

$$L = \text{Tr}(W^T VW) - 2\text{Tr}(U_p^T Z_p^T VZ_p U_p) + \text{Tr}(\Phi U_p)$$  \hspace{1cm} (10)

Then, let $L$ find the partial derivative of $U_p$, we can get:

$$\frac{\partial L}{\partial U_p} = -2(Z_p^T V W) + 2(Z_p^T VZ_p U_p) + \Phi$$  \hspace{1cm} (11)

Using the KKT condition ($\Phi_{ij} U_{pj} = 0$), we have:

$$-2(Z_p^T V W)_{ij} U_p{ij} + 2(Z_p^T VZ_p U_p)_{ij} U_p{ij} + \Phi_{ij} U_p{ij} = 0$$  \hspace{1cm} (12)

Subsequently, we can get the following updated iteration rules of $U_p$:

$$U_p{ij} \leftarrow U_p{ij} \frac{(Z_p^T V W)_{ij}}{(Z_p^T VZ_p U_p)_{ij}}$$  \hspace{1cm} (13)

**Algorithm 1** Algorithm to solve the RLSCC model of Eq (3)

**Require:** The original data $X \in \mathbb{R}^{N \times D}$, the number of anchors $M$, the Gaussian kernel bandwidth $\sigma$, and the number of $p$ columns of $Z$.

**Ensure:** The cluster indicator matrix $Y$.

1: Generate $M$ anchors from $N$ samples by using k-means.
2: Construct anchor graph by Eq (2).
3: Initialize $U_p$ to a random non-negative matrix, and get $Z_p$ from Eq (4).
4: while not converge do
5: Update $V$ by Eq (8).
6: Update $U_p$ by Eq (13).
7: end while

By iteratively updating $V$ and $U_p$ until the objection function converges, we can directly obtain the category to which each sample belongs from the optimal probabilistic clustering matrix $Y = Z_p U_p$. The details of the process are shown in Algorithm 1.

**Computational complexity**

The computational complexity of RLSCC is mainly composed of the following parts: anchors generation, anchor graph construction, and iterative optimization. The details of the complexity of these parts are as follows:

1. The complexity of $O(NMDT_1)$ is needed to use k-means to generate $M$ anchors from $N$ samples, and $T_1$ is the iteration number of k-means.
2. $O(NMD)$ complexity is desired when constructing anchor graph between $N$ samples and $M$ anchors by utilizing Eq 2.
3. $O((NC^2 + NCp)T)$ is needed when optimizing. To be precise, $O(NC^2)$ is needed to solve $V$, $O(NCp)$ is demanded to update $U_p$, and $T$ is the number of iteration of the objective function to convergence.

Generally speaking, the overall computational complexity of RLSCC is $O(NMDT_1 + (NC^2 + NCp)T)$. Since $M$, $D$, $p$, $C$, $T_1$, and $T$ are much smaller than $N$ when dealing with large-scale data, the complexity of RLSCC can be approximately $O(N)$. In particular, when the dimension of the data is large, RLSCC can still maintain a low computational complexity because it is...
independent of the dimension in the optimizing iteration part when solving the objective function.

**Experiments**

In this section, we give a comparison of the performance among RLSCC and six states-of-the-art algorithms (CF [45], LPFNMTF [16], LRS [46], LSSC [27], GCCF [33], EC [47]) on six datasets (TDT2 [48], Mnist [49], Corel [50, 51], Motper1 and Motper2 (http://www.escience.cn/people/fpnie/index.html), Corel [50, 51], and USPS [49]). Table 1 shows some properties of the six datasets. And also to validate the robustness of our methods, numerous experiments in noisy datasets are carried out. Specifically, six different metrics: ACC [52], NMI [53], Purity [54], ARI [55], F-score [56] and Precision [57] are used to verify the effectiveness and robustness of RLSCC.

### Compared methods and parameter setting

Six states-of-the-art clustering algorithms (CF, LPFNMTF, LRS, LSSC, GCCF, EC) are presented as compared methods in this part to verify the advantages of our algorithm over the mainstream clustering algorithms for large-scale data. A brief introduction of the comparison algorithms are outlined as follows:

- **CF**'s full name is concept factorization. It models each concept as a linear combination of the data points, and each data point as a linear combination of the concepts. Differing from the method of clustering based on non-negative matrix factorization (NMF) [58], this method can be applied to data with negative values and can be implemented in the kernel space.

- **LPFNMTF** is a local preserving regularization method based on fast non-negative matrix factorization. By using manifold regularization, this method can realize the geometric constraints on the two factorization factor matrices. What’s more, an optimization algorithm for LPFNMTF is proposed, which greatly improve efficiency by reducing the multiplication of matrix.

- **LRS** is a new subspace clustering model to cluster data which is drawn from multiple linear or affine subspaces. Instead of using two steps' algorithm (building the affinity matrix and spectral clustering). It directly learns the different subspaces’ indicator so that low-rank based different groups are obtained clearly. What’s more, this method use Schatten $p$-norm [59] to relax the rank constraint instead of using trace norm for better approximation of the low-rank constraint.

- **LSSC** is a large-scale sparse clustering algorithm, using $L_1$-norm for regularization to exploit matrix sparsity and obtain more robustness. Meanwhile, the model uses nonlinear approximation and dimension reduction techniques to further speed up the sparse coding algorithm, which brings high efficiency.

- **GCCF** is a clustering algorithm based on correntropy, which introduces the correntropy technique into the clustering analysis for the first time, and uses the correntropy to good suppression of nonlinear and non-Gaussian noise to improve the accuracy of clustering results.

---

**Table 1. Datasets description.**

| Datasets | Corel | Mnist | Motper1 | Motper2 | USPS | TDT2 |
|----------|-------|-------|---------|---------|------|------|
| Samples  | 5000  | 6996  | 5000    | 5000    | 9298 | 9394 |
| Dimension| 423   | 784   | 48      | 48      | 256  | 3677 |
| Classes  | 50    | 10    | 5       | 5       | 10   | 30   |
| Type     | Image | Image | Image   | Image   | Image | Text |

https://doi.org/10.1371/journal.pone.0277012.t001
EC’s full name is extreme clustering and it is a clustering method via density extreme points proposed for overcoming the drawbacks of peak clustering [60]. The theme of extreme clustering is to identify density extreme points to find cluster centers. What’s more, to guarantee the robustness, a noise detection module is also introduced to eliminate the influence of noisy data points.

In Table 2, we summarize the computational complexity of the compared methods. Some common notions for all methods, including the number of samples, classes, dimensions, and optimization iterations, are represented as $N, C, D, T$, respectively. Meanwhile, there are some method-specific notations whose meanings are as follows: $M_1$ in LPFNMTF indicates the additional dimension number introduced by NMTF, $M_2$, $p$, and $T_1$ in LSSC indicate the selected clustering centers for nonlinear approximation, the number of leading eigenvectors, and the iteration number of k-means, respectively.

For these compared methods which owns parameters (LPFNMTF, LSSC, and GCCF) affecting the clustering performance, our settings are as follows: LPFNMTF and GCCF own two parameters including the regularization parameter $\lambda$ and the number of nearest neighbors. For the two methods, we select $\lambda$ from the set {$1e1, 1e2, 1e3$} and $p$ from the set {3, 5, 7} to tune the results to the optimal results; For LSSC, the regularization parameter is set as 0.1 as author’s advice. All the compared methods are tune to their best based on our capability.

### Clustering results

In this part, we adopt six widely used metrics, which contain ACC, NMI, Purity, ARI, F-score, and Precision, to verify the performance of RLSCC and the compared methods on six datasets. For all clustering methods, the larger values of the metrics are expected to achieve better performance. To be fair, all experiments were performed five times on a laptop computer configured as a 16.0GB 3.20GHz AMD Ryzen CPU 5800H, at Matlab 2020b (64bit), and the mean values were recorded and the optimal and suboptimal results are marked in bold. Meanwhile, the mark and star indicate the computing time greater than 3 hours and the memory overflow when performing the experiment, respectively.

For the clustering efficiency, it can be observed from Table 2 in theory that compared to other methods, the complexity of the proposed method is less sensitive to the number of samples and the number of dimensions. And from the aspect of practice, Table 3 shows the computational time of various methods on different datasets, we can observe that RLSCC can achieve the same or better level of efficiency as high-efficient clustering methods like LPFNMTF, LSSC, and EC and hundreds of times faster than CF and LRS. What’s more, on

| Method      | Computational Complexity                       |
|-------------|-----------------------------------------------|
| CF          | $O(N^2 C T + N^{-1} D)$                       |
| LPFNMTF     | $O((N M_1 D + N M_2 C + N C^2 + M_1 C^2 + M_2 D + C^3 + M_3^2 T))$ |
| LRS         | $O(N C D^2 T)$                                |
| LSSC        | $O((N C D + N M_2 D) T_1 + N C^2 + (p^3 + N p^2 + N C p + C^3) T)$ |
| GCCF        | $O((N^2 C + N^2 D + N C) T)$                  |
| EC          | $O(N^2 D)$                                    |
| Ours        | $O(N M D T_1 + (N C^2 + N C p) T)$             |

$^a$ Here we only list the main complexity of EC since the uncertain complexity (but less than or equal to $O(N^2)$) in extreme points identification.

https://doi.org/10.1371/journal.pone.0277012.t002
TDT2, which is a high-dimensional data, the computational time of RLSCC is much more stable compared with these k-means based methods (LPFNMTF, GCCF, and CF), showing RLSCC is much more insensitive to data dimensions due to the implementation of pseudo-labels and graph learning. And compared with the robust methods (LRS, GCCF, and LSSC), especially when compared with GCCF which also uses correntropy to suppress noise, RLSCC shows more efficiency. The high efficiency of RLSCC benefits mainly from the pseudo-labels generation and anchors generation step which inherit the advantages of k-means and anchor-based anchor-based spectral clustering respectively. Concretely, the implementation of pseudo-labels and graph learning makes RLSCC insensitive to data dimensions while the strategies of anchor and directly obtaining the getting the sample classes further improve the efficiency.

As for the clustering effectiveness, Tables 4–6 show the effectiveness of RLSCC and the compared methods on six datasets. As presented in the tables, RLSCC can achieve the top two effectiveness in the six metrics and on six datasets. Especially, in some cases such as, on Corel,
Mnist, and USPS datasets, ACC, NMI, Purity, ARI, and F-score of RLSCC are averagely higher than the suboptimal results: 20.3%, 16.0%, 9.2%, 38.9%, 19.2%, and 33.4% respectively, which demonstrates the high effectiveness of RLSCC. When combining all the tables, it can be observed that RLSCC can improve the clustering efficiency greatly while guaranteeing comparable or even better clustering effectiveness.

Robustness analysis

As mentioned before, the introduction of correntropy can bring RLSCC resistance to various noises in real-world datasets. To verify the robustness of RLSCC, extensive experiments have been performed in eight noisy datasets. Specifically, we added different degrees (5%, 10%) of random noise and position noise to Corel and Mnist to form different noise datasets and performed RLSCC, and compared methods on these datasets under the same experimental conditions. The results are shown in Figs 1 and 2, from which we can obtain that the performance of RSCL can be maintained at the original level. Especially, compared with LSSC which uses the $L_1$-norm to achieve robustness, RLSCC gives better clustering performance and robustness in all cases when facing more complex (non-linear and non-Gaussian) noise, which shows the advantage of correntropy.

Parameter analysis

There are two main parameters contained in RLSCC: the number of anchors $M$, which affect the clustering effectiveness and efficiency, and the bandwidth of the Gaussian kernel $\delta$, which determines the robustness of the model. To validate the impact on the efficiency and effectiveness of these two parameters, we perform RLSCC under different parameter conditions and discuss the results in this part. The number of anchors has a huge effect on the clustering performance and efficiency. It is important to choose a suitable number of anchors to make a good trade-off between effectiveness and efficiency when performing RLSCC. To explore a proper $M$, extensive experiments were done using different $M$ from the set of $\{c+1, c+5, c+10, c+20, c+30, c+50\}$, where $c$ is the number of categories of the dataset. Fig 3 shows the clustering performance and computational time of different numbers of anchors when $\delta = 10$. On the one hand, the clustering performance shows an overall upward trend, and the upward trend is gradually becoming slower as the number of anchors increases. On the other hand, the computational time continues to increase with the growth of the number of anchors but it in general remains at a low level. Therefore, RLSCC can give a satisfying trade-off between efficiency and effectiveness via a suitable selection of the number of anchors. As another important parameter, the bandwidth of the Gaussian kernel $\delta$ impacts the robustness of RLSCC. Fig 4 presents the influence of different bandwidths of the Gaussian kernel when $M = 50$ on the final

| Method | USPS ACC | USPS NMI | USPS Purity | USPS ARI | USPS F-score | USPS Precision | TDT2 ACC | TDT2 NMI | TDT2 Purity | TDT2 ARI | TDT2 F-score | TDT2 Precision |
|--------|----------|---------|-------------|----------|---------------|----------------|----------|---------|-------------|----------|---------------|----------------|
| CF     | 0.3385   | 0.2464  | 0.3745      | 0.1617   | 0.2880        | 0.2115         | -        | -       | -           | -        | -             | -              |
| LPFNMFT| 0.1120   | 0.0020  | 0.0000      | 0.1036   | 0.1075        | *              | *        | *       | *           | *        | *             | *              |
| LRS    | 0.2206   | 0.0878  | 0.2629      | 0.0561   | 0.1920        | 0.1434         | *        | *       | *           | *        | *             | *              |
| LSSC   | 0.1670   | 0.0000  | 0.1670      | 0.1941   | 0.1075        | 0.1963         | 0.0000   | 0.0000  | 0.1963      | 0.0000   | 0.1974        | 0.1095         |
| GCCF   | 0.1671   | 0.0001  | 0.1672      | 0.1671   | 0.1941        | 0.1075         | *        | *       | *           | *        | *             | *              |
| EC     | 0.3319   | 0.4010  | 0.6049      | 0.1868   | 0.3153        | 0.2145         | 0.2385   | 0.0643  | 0.2471      | 0.0131   | 0.2052        | 0.1155         |
| Ours   | 0.6158   | 0.4503  | 0.6158      | 0.4184   | 0.4860        | 0.4496         | 0.5037   | 0.4093  | 0.5824      | 0.5171   | 0.4689        |                 |

https://doi.org/10.1371/journal.pone.0277012.t006
Fig 1. Clustering results of all algorithms with different noise on Corel. (a) ACC on Corel with different noise, (b) NMI on Corel with different noise, (c) Purity on Corel with different noise, (d) ARI on Corel with different noise, (e) F-score on Corel with different noise, and (f) Precision on Corel with different noise.

https://doi.org/10.1371/journal.pone.0277012.g001
Fig 2. Clustering results of all algorithms with different noise on Mnist. (a) ACC on Mnist with different noise, (b) NMI on Mnist with different noise, (c) Purity on Mnist with different noise, (d) ARI on Mnist with different noise, (e) F-score on Mnist with different noise, and (f) Precision on Mnist with different noise.

https://doi.org/10.1371/journal.pone.0277012.g002
clustering results and computational time from an experimental point of view. In these experiments, $\delta$ is selected in $\{1, 10, 50, 100, 500, 1000\}$. We can obtain that the clustering results and computational time basically hover in a certain and acceptable range with the increase of $\delta$.

**Conclusion**

This paper proposes a robust large-scale clustering algorithm based on correntropy (RLSCC), which inherits the low computational complexity of k-means and the high effectiveness of
spectral clustering. Meanwhile, the generation of pseudo-labels and the use of anchor graphs can effectively improve the efficiency of clustering. To solve RLSSC, a new fast optimization algorithm based on half-quadratic technology is proposed, which can complete the confirmation of the sample category in a short time. Finally, extensive experiments on real-world datasets and noisy datasets show that compared to other state-of-the-art algorithms, especially when facing large-scale high-dimensional data, RLSCC can ensure higher efficiency and robustness while remaining comparable or even better clustering effectiveness. However, there are still some limitations to the present method. On the one hand, the performance of k-means is easily affected by initialization, which may affect the generation quality of pseudo-labels and anchor graphs and further affect the clustering effectiveness. On the other hand, the proposed method cannot be applied to multi-view datasets, which are now common in real applications.

Fig 4. Clustering results of RLSCC with different \( \delta \) on Corel and Mnist. (a) Clustering results on Corel and (b) Clustering results on Mnist.

https://doi.org/10.1371/journal.pone.0277012.g004
Therefore, the future scope of the present work is to apply novel methods for pseudo-labels and anchor graph generation and to extend the work to a multi-view version.

**Author Contributions**

**Conceptualization:** Guodong Jin.

**Data curation:** Lining Tan.

**Formal analysis:** Jing Gao.

**Funding acquisition:** Jing Gao.

**Investigation:** Jing Gao.

**Methodology:** Guodong Jin.

**Software:** Lining Tan.

**Visualization:** Lining Tan.

**Writing – original draft:** Guodong Jin.

**Writing – review & editing:** Jing Gao.

**References**

1. Razzak MI, Naz S, Zaib A. Deep learning for medical image processing: Overview, challenges and the future Classification in BioApps. 2018;323–350.
2. Jiao L, Zhao J. A survey on the new generation of deep learning in image processing. IEEE Access. 2019; 7:172231–172263. https://doi.org/10.1109/ACCESS.2019.2956508
3. Jiao L, Zhao J, Feng SJ, Yin W, Li YX, Fan PF, et al. Deep learning in optical metrology: a review. Light: Science & Applications. 2022; 11(1):1–54.
4. Suganyadevi S, Seethalakshmi V, Balasamy K. A review on deep learning in medical image analysis. International Journal of Multimedia Information Retrieval. 2022; 11(1):19–38. https://doi.org/10.1007/s13735-021-00218-1 PMID: 34513553
5. Karanam SR, Srinivas Y, Krishna MV. Study on image processing using deep learning techniques. Materials Today: Proceedings. 2020.
6. Haq MA. Planetscope Nanosatellites Image Classification Using Machine Learning. Computer System Science and Engineering. 2022; 42(3):1031–1046. https://doi.org/10.32604/csse.2022.023221
7. Haq MA. CNN Based Automated Weed Detection System Using UAV Imagery. Computer System Science and Engineering. 2022; 42(2):837–849. https://doi.org/10.32604/csse.2022.023016
8. Haq MA. Smotednn: A novel model for air pollution forecasting and aqi classification. Computers, Materials and Continua. 2022; 71:1.
9. Haq MA. CDLSTM: A novel model for climate change forecasting. Computers, Materials and Continua. 2022; 71:2363–2381. https://doi.org/10.32604/cmc.2022.023059
10. Haq MA, Prabu P. Deep Learning Based Modeling of Groundwater Storage Change. Computers, Materials and Continua. 2021; 70:4599–4617.
11. Haq MA, Rahaman G, Baral P, Ghosh A. Deep learning based supervised image classification using UAV images for forest areas classification. Journal of the Indian Society of Remote Sensing. 2021; 49 (3):601–606. https://doi.org/10.1007/s12524-020-01231-3
12. Haq MA, Baral P, Yaragal S, Pradhank B. Bulk Processing of Multi-Temporal Modis Data, Statistical Analyses and Machine Learning Algorithms to Understand Climate Variables in the Indian Himalayan Region. Sensors. 2021; 21(21):7416. https://doi.org/10.3390/s21217416 PMID: 34770722
13. Haq MA, Baral P. Study of permafrost distribution in Sikkim Himalayas using Sentinel-2 satellite images and logistic regression modelling Geomorphology. 2019; 333:123–136.
14. Haq MA, Azam MF, Vincent C. Efficiency of artificial neural networks for glacier ice-thickness estimation: A case study in western Himalaya, India Journal of Glaciology. 2021; 67(264):671–684.
15. Nie F, Wang CL, Li X. K-multiple-means:A multiple-means clustering method with specified k clusters. Association for Computing Machinery. 2019:959–967.
16. Wang H, Nie F, Huang H, Makedon F. Fast nonnegative matrix tri-factorization for large-scale data Co-Clustering. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. 2011:1553-1558

17. Han J, Song K, Nie F, Li X. Bilateral k-Means algorithm for fast co-clustering. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017:1969-1975.

18. Zhang R, Rudnicky AI. A large scale clustering scheme for kernel k-means. Object recognition supported by user interaction for service robots. 2002; 4:289-292

19. Yang B and Li Z, Zhang X, Nie F, Wang F. Efficient Multi-view K-means Clustering with Multiple Anchor Graphs. IEEE Transactions on Knowledge and Data Engineering. 2022.

20. Nie F, Zhu W, Li X. Unsupervised large graph embedding. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017:2422-2428.

21. Wang R, Nie F, Yu W. Fast spectral clustering with anchor graph for large hyperspectral images. IEEE Geoscience and Remote Sensing Letters. 2017; 14 (11):2003–2007. https://doi.org/10.1109/LGRS.2017.2746625

22. Yang X, Yu W, Wang R, Zhang G, Nie F. Fast spectral clustering learning with hierarchical bipartite graph for large-scale data. Pattern Recognition Letters. 2020; 130:345–352. https://doi.org/10.1016/j.patrec.2018.06.024

23. Wang CL, Nie F, Wang R, Li X. Revisiting fast spectral clustering with anchor graph. IEEE International Conference on Acoustics, Speech and Signal Processing. 2020:3902-3906.

24. Zhu W, Nie F, Li X. Fast spectral clustering with efficient large graph construction. IEEE International Conference on Acoustics, Speech and Signal Processing. 2017:2492-2496.

25. Yang B, Zhang X, Nie F, Wang F. Fast Multi-view Clustering with Spectral Embedding IEEE Transactions on Image Processing. 2022. PMID: 35609096

26. Yang B, Zhang X, Nie F, Wang F, Yu W, Wang R. Fast multi-view clustering via nonnegative and orthogonal factorization. IEEE Transactions on Image Processing. 2021; 30:2575–2586. https://doi.org/10.1109/TIP.2020.3049631 PMID: 33360992

27. Zhang R, Lu Z. Large scale sparse clustering. Proceedings of theTwenty-Fifth International Joint Conference on Artificial Intelligence. 2016:2336-2342.

28. Guo Y, Ding G, Zhou J, Liu Q. Robust and discriminative concept factorization for image representation. Proceedings of the fifth ACM on International Conference on Multimedia Retrieval. 2015:115–122.

29. Zhu X, Zhang S, Li Y, Zhang J, Yang L, Fang Y. Low-rank sparse subspace for spectral clustering. IEEE Transactions on Knowledge and Data Engineering, 2019; 31 (8):1532–1543. https://doi.org/10.1109/TKDE.2018.2858782

30. Liu G, Lin Z, Yu Y. Robust subspace segmentation by low-rank representation. Proceedings of theTwenty-sixth International Conference on Machine Learning. 2010.

31. Yang B, Wu J, Sun A, Gao N, Zhang X. Robust landmark graph-based clustering for high-dimensional data. Neurocomputing. 2022; 496:72–84. https://doi.org/10.1016/j.neucom.2022.05.011

32. Principe JC. Information theoretic learning:Renyi's entropy and kernel perspectives. Springer Science and Business Media, 2010.

33. Yang B and Zhang X, Lin Z, Nie F, Chen B, Wang F. Efficient and Robust Multi-view Clustering with Anchor Graph Regularization. IEEE Transactions on Circuits and Systems for Video Technology. 2022.

34. Peng S, Ser W, Chen B, Sun L, Lin Z. Correntropy based graph regularized concept factorization for clustering. Neurocomputing. 2018; 316:34–48. https://doi.org/10.1016/j.neucom.2018.07.049

35. Yu N, Wu M, Liu J, Zheng C, Xu Y. Correntropy-based hypergraph regularized NMF for clustering and feature selection on multi-cancer integrated data. IEEE Transactions on Cybernetics. 2021; 51 (8):3952–3963. https://doi.org/10.1109/TCYB.2020.3000799 PMID: 32603306

36. Peng S, Ser W, Chen B, Lin Z. Robust semi-supervised nonnegative matrix factorization for image clustering. Pattern Recognition. 2021; 111:107683. https://doi.org/10.1016/j.patcog.2020.107683

37. Yang B, Zhang X, Nie F, Chen B, Wang F, Nan Z, et al. ECCA: Efficient Correntropy-Based Clustering Algorithm With Orthogonal Concept Factorization. IEEE Transactions on Neural Networks and Learning Systems. 2022. https://doi.org/10.1109/TNNLS.2022.3142806 PMID: 35100124

38. Yang B, Zhang X, Chen B, Nie F, Lin Z, Nan Z. Efficient correntropy-based multi-view clustering with anchor graph embedding. Neural Networks. 2022; 146:290–302 https://doi.org/10.1016/j.neunet.2021.11.027 PMID: 34915413

39. Zhou N, Xu Y, Cheng H, Yuan Z, Chen B. Maximum correntropy criterion-based sparse subspace learning for unsupervised feature selection. IEEE Transactions on Circuits and Systems for Video Technology. 2017; 29(2):404–417. https://doi.org/10.1109/TCSVT.2017.2783364
40. Geman D, Reynolds G. Constrained restoration and the recovery of discontinuities. IEEE Transactions on pattern analysis and machine intelligence. 1992; 14(2):367–383. https://doi.org/10.1109/34.120331
41. He R., Zheng W. S., Tan T., Sun Z., Half-quadratic-based iterative minimization for robust sparse representation, IEEE transactions on pattern analysis and machine intelligence, 36 (2) (2013) 261–275.
42. Liu J, Han J. Spectral clustering, Data Clustering, Chapman and Hall/CRC, (2018) 177–200.
43. Nie F, X. Wang, M. Jordan, Huang H. The constrained Laplacian rank algorithm for graph-based clustering, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, (2016) 1969-1976.
44. C. Wang, Nie F, Wang R, Li X, Revisiting fast spectral clustering with anchor graph, Proceedings of the Forty-fifth International Conference on Acoustics, Speech, and Signal Processing, (2020) 3902-3906.
45. Xu W, Gong Y. Document clustering by concept factorization. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 2004:202-209.
46. Nie F, Huang H. Subspace clustering via new low-rank model with discrete group structure constraint. International Joint Conference on Artificial Intelligence. 2016:1874-1880.
47. Wang S, Li Q, Zhao C, Zhu X, Yuan H, Dai T. Extreme clustering—a clustering method via density extreme points. Information Sciences. 2021; 542:24–39. https://doi.org/10.1016/j.ins.2020.06.069
48. Fiscus J, Dodgington G, Garofolo J, Martin A. Nist’s 1998 topic detection and tracking evaluation (tdt2). Proceedings of the 1999 DARPA Broadcast News Workshop. 1999:19–24.
49. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 6(11):2288–2324. https://doi.org/10.1109/5.726791
50. Barnard K, Johnson M. Word sense disambiguation with pictures. Artificial Intelligence. 2005; 167(1-2):13–30. https://doi.org/10.1016/j.artint.2005.04.009
51. Barnard K, Duygulu P, Forsyth D, Freitas N De, Blei DM, Jordan M. Matching words and pictures. 2003.
52. Wu M, Schölkopf B. A local learning approach for clustering. Advances in neural information processing systems. 2006; 19:1529–1536.
53. Ana LF, Jain AK. Robust data clustering. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2003:2-II.
54. Schütze H, Manning CD, Raghavan P. Introduction to information retrieval. 39.
55. Steinley D. Properties of the hubert-arable adjusted rand index. Psychological methods. 2004; 9(3):86. https://doi.org/10.1037/1082-989X.9.3.386 PMID: 15355155
56. Sokolova M, Japkowicz N, Szpakowicz S. Beyond accuracy, f-score and roc-a family of discriminant measures for performance evaluation. Australasian joint conference on Artificial intelligence. 2006:1015-1021.
57. Powers DM. Recall and precision versus the bookmaker. International Conference on Cognitive Science. 2003.
58. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999; 401(6755):788–791. https://doi.org/10.1038/44565 PMID: 10548103
59. Nie F, Huang H, Ding C. Low-rank matrix recovery via efficient schatten p-norm minimization. Twenty-sixth AAAI conference on artificial intelligence. 2012.
60. Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science. 2014; 34 (6191):1492–1496. https://doi.org/10.1126/science.1242072 PMID: 24970081