Tri-level Robust Clustering Ensemble with Multiple Graph Learning

Peng Zhou, [1 2] **Liang Du,** [3] **Yi-Dong Shen,** [2*] **Xuejun Li** [1]

[1]School of Computer Science and Technology, Anhui University
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[3]School of Computer and Information Technology, Shanxi University
zhoupeng@ahu.edu.cn, duliang@sxu.edu.cn, ydshen@ios.ac.cn, xjli@ahu.edu.cn

Abstract

Clustering ensemble generates a consensus clustering result by integrating multiple weak base clustering results. Although it often provides more robust results compared with single clustering methods, it still suffers from the robustness problem if it does not treat the unreliability of base results carefully. Conventional clustering ensemble methods often use all data for ensemble, while ignoring the noises or outliers on the data. Although some robust clustering ensemble methods are proposed, which extract the noises on the data, they still characterize the robustness in a single level, and thus they cannot comprehensively handle the complicated robustness problem. In this paper, to address this problem, we propose a novel Tri-level Robust Clustering Ensemble (TRCE) method by transforming the clustering ensemble problem to a multiple graph learning problem. Just as its name implies, the proposed method tackles robustness problem in three levels: base clustering level, graph level and instance level. By considering the robustness problem in a more comprehensive way, the proposed TRCE can achieve a more robust consensus clustering result. Experimental results on benchmark datasets also demonstrate it. Our method often outperforms other state-of-the-art clustering ensemble methods. Even compared with the robust ensemble methods, ours also performs better.

Introduction

Clustering ensemble provides an elegant framework to ensemble multiple weak base clustering results to generate a consensus one, and attracts much attention in unsupervised learning (Strehl and Ghosh 2003; Li, Ding, and Jordan 2007; Yu et al. 2015; Liu et al. 2015, 2018; Wang et al. 2019; Li et al. 2019; Zhou et al. 2020; Kang et al. 2020b; Zhou, Du, and Li 2020; Bai, Liang, and Cao 2020). For example, (Li, Ding, and Jordan 2007) applied nonnegative matrix factorization to learn a consensus clustering result; (Liu et al. 2018) developed a de-noising auto-encoder for clustering ensemble; (Li et al. 2019) used the base clusterings to evaluate the stability of each instance and assigned all the instances to some stable instances to form a cluster. Although these clustering ensemble methods can alleviate the robustness problem in single clustering methods to some extent, since they use all data for ensemble while ignoring the noises on the data, they still suffer from the robustness problem.

To address this issue, some robust clustering ensemble methods are proposed (Zhou et al. 2015a; Huang, Lai, and Wang 2016; Tao et al. 2017, 2019). However, these methods just focus on one single level of robustness. For example, (Zhou et al. 2015a; Tao et al. 2017, 2019) concentrated on the noises on the connective matrices; (Huang, Lai, and Wang 2016) paid attention on the graph level of robustness. Unfortunately, in clustering ensemble, the noises are far more complicated. The noises may exist in the original data, i.e., some instances are contaminated by noises or outliers. The noises may also be introduced when using inappropriate clustering methods. For example, we use k-means to handle non-linear data, which may achieve poor base results. Therefore, it is not enough to characterize the robustness in one single level. Another problem of these robust methods is that they only focus on contaminated data, or in other words, they completely trust the uncontaminated data. However, even for the uncontaminated data, some data are easy to handle and some are difficult which need to be handled more carefully. For example, the data in the boundary of two clusters, are difficult to assign to a cluster. If we deal with these difficult data too early, the early model may have not such ability to handle these difficult data and is easy to be misled by them.

To tackle these two problems, in this paper, we propose a novel Tri-level Robust Clustering Ensemble (TRCE) method. Since the clustering result reflects the relationship between instances, we construct a graph for each base clustering result to represent such relationship. In the graph, each node represents an instance, and if two instances belong to the same cluster, there exists an edge between the two corresponding nodes. Then, we tackle the noises on these multiple graphs. Just as its name implies, TRCE deals with noises on three levels: base clustering level, graph level, and instance level. Firstly, as introduced before, since inappropriate clustering methods may introduce noises, the quality of base clusterings are different. In the base clustering level of robustness, we will evaluate the quality of each base clustering, and aim to alleviate the side effects caused by the bad base clusterings. Secondly, since the multiple graphs are constructed by unreliable base clusterings, the graphs are also contaminated by noises (Kang et al. 2020a). To handle

*Corresponding author.

these graph level noises, we explicitly extract noises from graphs to recover a clean consensus graph. Last but not least, the original instances may also be contaminated by noises and outliers, we apply a self-paced learning (Kumar, Packer, and Koller 2010) to achieve this instance level of robustness. The self-paced learning automatically and incrementally use data for learning, where easy data are used first and difficult ones are then involved gradually (Jiang et al. 2014; Meng, Zhao, and Jiang 2017; Zhang, Meng, and Han 2017). Thanks to the self-paced learning, our TRCE can naturally tackle the second problem introduced before. Note that, the noisy data or outliers can be regarded as the most difficult data in the learning, because they may contribute nothing for the learning. Besides, the uncontaminated data can also be handled in order of their difficulty. Then, the difficult data, which may mislead the model, will be involved in learning until the model has the ability to handle them. Therefore, our TRCE handles all data in a more sophisticated way.

To fulfil this tri-level robust ensemble method, we carefully design a unified objective function to handle the three levels of robustness simultaneously. Then, we provide an effective iterative algorithm to optimize the introduced objective function. The algorithm decomposes the objective function into several subproblems, and finds the global optima for each subproblem. At last, we conduct extensive experiments on benchmark datasets to compare TRCE with state-of-the-art clustering ensemble methods. The experimental results show that TRCE can outperform not only the conventional clustering ensemble methods but also the state-of-the-art robust clustering ensemble methods.

It is worthy to highlight the main contributions of this paper as follows:

- We propose a novel robust clustering ensemble method which tackles the robustness problem in three levels. It provides a more comprehensive way to deal with noises than existing robust ensemble methods.
- Different from existing robust clustering ensemble methods, which only pay attention on the contaminated data, we plug the robust clustering ensemble into a self-paced multiple graph learning framework. Due to the self-paced learning schema, the proposed method can appropriately handle all contaminated and uncontaminated data.
- We integrate the tri-level robust clustering ensemble and the self-paced multiple graph learning into a unified objective function, and designed an iterative algorithm to optimize it. In our optimization algorithm, each subproblem can be solved by finding its global optima. We obtain the final clustering result in an end-to-end way without any uncertain postprocessing.
- The experimental results show that the proposed method outperforms the state-of-the-art clustering ensemble methods, including the robust ones.

Tri-level Robust Clustering Ensemble

Before introducing TRCE in detail, we introduce some notations. Throughout the paper, the matrices and vectors are denoted by boldface uppercase and lowercase letters, respectively. For a matrix $\mathbf{A}$, $\mathbf{A}_{i\cdot}$ and $\mathbf{A}_{\cdot i}$ represents the i-th row and column vector of $\mathbf{A}$, respectively. The (i,j)-th element of $\mathbf{A}$ is denoted as A_{ij}.

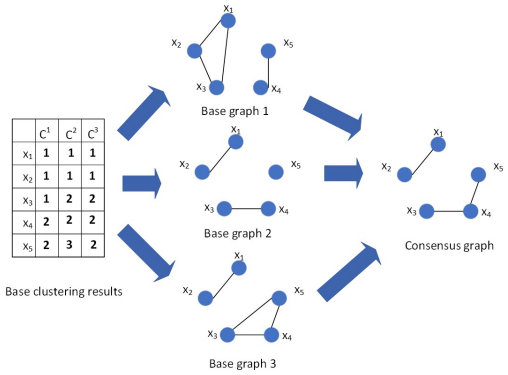


Figure 1: An illustration of constructing multiple graphs.

Given m base clustering results, we construct m graphs based on them. In the k-th graph, there is an edge between the i-th and j-th node if the i-th instance and j-th instance belongs to the same cluster in the k-th base clustering, and there is no edge if otherwise. Figure 1 shows an example. In this example, we have 5 instances (denoted as $\mathbf{x}_1, \cdots, \mathbf{x}_5$) and 3 base clusterings (denoted as $\mathbf{C}^1, \cdots, \mathbf{C}^3$). In the first base clustering result, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ belong to the first cluster, and $\mathbf{x}_4$ and $\mathbf{x}_5$ belong to the second cluster. The 3 base graphs constructed by the 3 base clustering results are shown in the middle of Figure 1. The goal of our multiple graph learning is to learn a consensus graph (shown in the right of Figure 1) by ensembling the multiple base graphs.

More formally, we use $\mathbf{S}^{(1)}, \cdots, \mathbf{S}^{(m)} \in \{0,1\}^{n\times n}$ to denote the adjacent matrices of the m graphs, i.e., $S_{ij}^{(k)} = 1$ if there exists an edge between the i-th and j-th instances in the k-th graph, and $S_{ij}^{(k)} = 0$ otherwise. Then we normalize them by $\mathbf{A}^{(k)} = (\mathbf{D}^{(k)})^{-1}\mathbf{S}^{(k)}$ as probability transition matrices, where $\mathbf{D}^{(k)}$ is a diagonal matrix whose diagonal elements are $D_{ii}^{(k)} = \sum_{j=1}^{n} S_{ij}^{(k)}$. It is easy to verify that $\sum_{j=1}^{n} \mathbf{A}_{ij}^{(k)} = 1$, and $A_{ij}^{(k)}$ indicates the probability of jumping in one step from the i-th instance to the j-th instance in a Markov random walk in the k-th graph. To integrate the multiple graphs, we need to learn a consensus transition matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ from $\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(m)}$. In the following subsections, we will introduce how to learn the consensus transition matrix $\mathbf{A}$ by considering the three levels of robustness.

Base Clustering Level of Robustness

A natural way to learn $\mathbf{A}$ is to minimize the difference between $\mathbf{A}$ and $\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(m)}$. Considering that the elements in a row of transition matrix has a clear probabilistic interpretation, we use the Kullback-Leibler divergence to measure the differences. More formally, we minimize $\sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n} \mathrm{KL}(\mathbf{A}_{i\cdot}^{(k)}, \mathbf{A}_{i\cdot})$, where $\mathrm{KL}(\cdot,\cdot)$ is the Kullback-Leibler divergence. α_k is the non-negative weight of the k-th graph which controls the base clustering level of robustness, i.e., the worse base clusterings, which are far

away from the consensus one, will have smaller weights.

Moreover, to learn a consensus clustering result from $\mathbf{A}$ without any postprocessing which may introduce uncertainty and randomness, we wish there are just c connective components in $\mathbf{A}$. To achieve this, according to (Nie, Wang, and Huang 2014), we have that the number of connective components in $\mathbf{A}$ is equal to the multiplicity of the eigenvalue 0 of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \frac{\mathbf{A}+\mathbf{A}^T}{2}$, where $\mathbf{D}$ is a diagonal matrix whose i-th diagonal element is $D_{ii} = \sum_{j=1}^{n} \frac{A_{ij}+A_{ji}}{2}$. Thus, we need to impose the constraint $rank(\mathbf{L}) = n - c$ on $\mathbf{A}$. To sum up, we take the definition of Kullback-Leibler divergence into it, and obtain the following objective function

$$\begin{aligned} \min_{\mathbf{A},\boldsymbol{\alpha}} \quad & \sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^{(k)} \log\left(\frac{A_{ij}^{(k)}}{A_{ij}}\right) \qquad (1) \\ s.t. \quad & \sum_{j=1}^{n} A_{ij} = 1, \quad 0 \le A_{ij} \le 1, \\ & \sum_{k=1}^{m} \alpha_k^{-1} = 1, \quad \alpha_k \ge 0, \quad rank(\mathbf{L}) = n - c, \end{aligned}$$

where the constraint $\sum_{k=1}^{m} \alpha_k^{-1} = 1$ is to avoid the trivial solution.

Graph Level of Robustness

From the robustness point of view, we observe that $\mathbf{A}$ learned by Eq.(1) often contains some noises due to the unreliability of the base graphs. For example, considering the case that for the i-th and j-th instances, only one graph shows that they should be linked by an edge, and the other $m-1$ graphs agree that they should not have an edge. Without loss of generality, we assume that $A_{ij}^{(1)} > 0$ and all other $A_{ij}^{(k)} = 0$ $(k = 2, \cdots, m)$. Obviously, in the consensus graph $\mathbf{A}$, the i-th and j-th instances are very likely not to have a connection, i.e., A_{ij} should be zero. However, by minimizing Eq.(1), A_{ij} should be non-zero. If not, considering the first term $A_{ij}^{(1)} \log\left(\frac{A_{ij}^{(1)}}{A_{ij}}\right)$, since $A_{ij}^{(1)} > 0$, if $A_{ij} = 0$, the objective will be infinity large. Thus, this term prevents A_{ij} being zero, and $\mathbf{A}$ may be very dense.

To characterize the noises on $\mathbf{A}$, we define a zero-mean noise matrix $\mathbf{E} \in \mathbb{R}^{n \times n}$ and apply $\mathbf{A} + \mathbf{E}$ to represent the consensus transition matrix and thus $\mathbf{A}$ denotes the clean and sparse consensus matrix. To this end, we extend Eq.(1) to the following form:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{E},\boldsymbol{\alpha}} & \sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^{(k)} \log\left(\frac{A_{ij}^{(k)}}{A_{ij}+E_{ij}}\right) + \lambda \|\mathbf{E}\|_F^2 \\ s.t. & \sum_{j=1}^{n} A_{ij} = 1, \quad 0 \le A_{ij} \le 1, \\ & \sum_{k=1}^{m} \alpha_k^{-1} = 1, \quad \alpha_k \ge 0, \quad rank(\mathbf{L}) = n - c, \\ & \sum_{j=1}^{n} E_{ij} = 0, \quad 0 \le A_{ij} + E_{ij} \le 1, \qquad (2) \end{aligned}$$

where $\|\mathbf{E}\|_F^2$ is the regularized term which imposes the prior that noises should be as small as possible and λ is a hyper-parameter to control the noises. The constraints on $\mathbf{E}$ makes sure that $\mathbf{A} + \mathbf{E}$ is a legal probability transition matrix, i.e., $\sum_{j=1}^{n} A_{ij} + E_{ij} = 1$ and $0 \le A_{ij} + E_{ij} \le 1$.

By introducing the noise matrix $\mathbf{E}$, we explicitly extract the noises on the consensus graph, which achieves the graph level of robustness.

Instance Level of Robustness

Besides the noises on the graph, the uncontaminated instances should also be handled carefully. Some instances are difficult for clustering and may mislead the model in beginning of learning, because the early model may not have the ability to handle them.

To address this issue, we seamlessly integrate the robust multiple graph learning into a self-paced learning framework, i.e., we gradually involve instances from easy ones to difficult ones into ensemble learning. Intuitively, for an easy instance $\mathbf{x}_i$, the m base graphs should agree with each other on its transition probability of jumping to other instances, i.e., $\sum_{k=1}^{m} \alpha_k \mathrm{KL}(\mathbf{A}_{i.}^{(k)}, \mathbf{A}_{i.}+\mathbf{E}_{i.})$ should be as small as possible. To characterize the difficulty of each instance, we introduce a weight vector $\mathbf{w} \in [0,1]^n$, i.e., the larger w_i is, the easier the i-th instance is. By introducing the self-paced regularized term $-\gamma\|\mathbf{w}\|_1$ as suggested in (Kumar, Packer, and Koller 2010; Jiang et al. 2015; Ren et al. 2019), we obtain the final objective function:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{E},\mathbf{w},\boldsymbol{\alpha}} & \sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n} w_i^2 \sum_{j=1}^{n} A_{ij}^{(k)} \log\left(\frac{A_{ij}^{(k)}}{A_{ij}+E_{ij}}\right) \\ & + \lambda\|\mathbf{E}\|_F^2 - \gamma\|\mathbf{w}\|_1 \\ s.t. & \sum_{j=1}^{n} A_{ij} = 1, \quad 0 \le A_{ij} \le 1, \\ & \sum_{k=1}^{m} \alpha_k^{-1} = 1, \quad \alpha_k \ge 0, \quad rank(\mathbf{L}) = n - c, \\ & \sum_{j=1}^{n} E_{ij} = 0, \quad 0 \le A_{ij} + E_{ij} \le 1, \\ & 0 \le w_i \le 1, \qquad (3) \end{aligned}$$

where γ is an age parameter and becomes increasingly larger in the process of optimization. By minimizing Eq.(3), we handle all instances (no matter contaminated ones or uncontaminated ones) carefully. The contaminated instances can be regarded as the most difficult or unreliable data, which may mislead the model, and will not be involved in learning at the beginning. Even for the uncontaminated instances, they are used in learning in order of their difficulty. Therefore, this self-paced learning framework provides a more sophisticated way to handle all instances than the conventional robust methods.

Eq.(3) is our final objective function. It has two characters: 1) it is a unified objective function which simultaneously controls the three levels of robustness; 2) it obtains the final consensus clustering results in an end-to-end way, i.e., it does not need any uncertain postprocessing such as k-means or spectral clustering.

Optimization

Since Eq.(3) contains the constraint $rank(\mathbf{L}) = n - c$ which is hard to deal with directly, we first rewrite this constraint.

According to Ky Fan Theorem (Fan 1949), by introducing an auxiliary orthogonal matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$ and a large enough parameter ρ, Eq.(3) can be equivalently rewritten as:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{E},\mathbf{w},\mathbf{F},\boldsymbol{\alpha}} \quad & \sum_{k=1}^{m}\alpha_k\sum_{i=1}^{n}w_i^2\sum_{j=1}^{n}A_{ij}^{(k)}\log\left(\frac{A_{ij}^{(k)}}{A_{ij}+E_{ij}}\right)\\ & +\lambda\|\mathbf{E}\|_F^2-\gamma\|\mathbf{w}\|_1+2\rho tr(\mathbf{F}^T\mathbf{L}\mathbf{F})\\ s.t. \quad & \sum\nolimits_{j=1}^{n}A_{ij}=1,\quad 0\le A_{ij}\le 1,\\ & \sum\nolimits_{k=1}^{m}\alpha_k^{-1}=1,\quad \alpha_k\ge 0,\\ & \sum\nolimits_{j=1}^{n}E_{ij}=0,\quad 0\le A_{ij}+E_{ij}\le 1\\ & 0\le w_i\le 1,\quad \mathbf{F}^T\mathbf{F}=\mathbf{I}. \end{aligned} \tag{4}$$

Then we will optimize each variable in Eq.(4) while fixing the other variables.

Optimizing w When optimizing $\mathbf{w}$, we find that the objective function can be de-coupled into n independent sub-problems. Considering the i-th one:

$$\begin{aligned}\min_{w_i} \quad & w_i^2 b_i-\gamma w_i\\ s.t. \quad & 0\le w_i\le 1. \end{aligned} \tag{5}$$

where $b_i=\sum_{k=1}^{m}\alpha_k\sum_{j=1}^{n}A_{ij}^{(k)}\log\left(\frac{A_{ij}^{(k)}}{A_{ij}+E_{ij}}\right)$. By setting the derivative w.r.t w_i to zero, we obtain $w_i=\frac{\gamma}{2b_i}$. Since $b_i\ge 0$, $w_i\ge 0$. If $\frac{\gamma}{2b_i}>1$, Eq.(5) decreases monotonically in the range $[0,1]$, and thus the optima is 1. Therefore, the closed form solution of Eq.(5) is

$$w_i=\min\left(\frac{\gamma}{2b_i},1\right) \tag{6}$$

Note that large b_i means that the multiple graphs hardly agree with each other on the distribution of the transition probability of the i-th instance, i.e., the i-th instance is difficult. Since w_i is proportional to $\frac{1}{b_i}$, large b_i will lead to small w_i, which means difficult instances will hardly influence the model. Moreover, γ plays a role as an age parameter. When γ grows, w_i also grows, i.e., more and more instances will be involved in learning. This reflects the instance level of robustness.

Optimizing A and E Since $\mathbf{A}$ and $\mathbf{E}$ are entangled in both the objective function and the constraints, for simplicity, we introduce an auxiliary variable $\mathbf{B}=\mathbf{A}+\mathbf{E}$. Note that $2tr(\mathbf{F}^T\mathbf{L}\mathbf{F})=\sum_{i,j=1}^{n}\|\mathbf{F}_{i.}-\mathbf{F}_{j.}\|_2^2A_{ij}=\sum_{i,j=1}^{n}\|\mathbf{F}_{i.}-\mathbf{F}_{j.}\|_2^2(B_{ij}-E_{ij})$. Thus, we can decompose Eq.(4) into the following two sub-problems w.r.t. $\mathbf{B}$ and $\mathbf{E}$, respectively.

$$\begin{aligned}\min_{\mathbf{B}} \quad & -\sum_{i=1}^{n}\sum_{j=1}^{n}H_{ij}\log(B_{ij})+\rho\sum_{i=1}^{n}\sum_{j=1}^{n}G_{ij}B_{ij}\\ s.t. \quad & 0\le B_{ij}\le 1,\quad \sum\nolimits_{j=1}^{n}B_{ij}=1. \end{aligned} \tag{7}$$

and

$$\begin{aligned}\min_{\mathbf{E}} \quad & \gamma\|\mathbf{E}\|_F^2-\rho\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n}G_{ij}E_{ij}\\ s.t. \quad & B_{ij}-1\le E_{ij}\le B_{ij},\quad \sum\nolimits_{j=1}^{n}E_{ij}=0. \end{aligned} \tag{8}$$

where $H_{ij}=\sum_{k=1}^{m}\alpha_k w_i^2A_{ij}^{(k)}$ and $G_{ij}=\|\mathbf{F}_{i.}-\mathbf{F}_{j.}\|_2^2$. Then we solve Eqs.(7) and (8) respectively.

When solving $\mathbf{B}$, Eq.(7) can be de-coupled into n independent sub-problems by rows. Considering the i-th one, note that due to the constraints $B_{ij}\ge 0$ and $\sum_{j=1}^{n}B_{ij}=1$, the constraint $B_{ij}\le 1$ can be dropped safely. Then, by introducing the Lagrange multipliers θ and μ_j, we obatin its Lagrange function:

$$\begin{aligned}\mathcal{L}=&-\sum\nolimits_{j=1}^{n}H_{ij}\log(B_{ij})+\rho\sum\nolimits_{j=1}^{n}G_{ij}B_{ij}\\ &+\theta(\sum\nolimits_{j=1}^{n}B_{ij}-1)-\sum\nolimits_{j=1}^{n}\mu_jB_{ij}\end{aligned} \tag{9}$$

Setting the partial derivative of $\mathcal{L}$ w.r.t. B_{ij} to zero, we get:

$$\frac{\partial\mathcal{L}}{\partial B_{ij}}=-\frac{H_{ij}}{B_{ij}}+\rho G_{ij}+\theta-\mu_j=0.$$

Since Eq.(7) is convex, and has a lower bound, a solution which satisfies the Karush-Kuhn-Tucker (KKT) conditions is the global optima of this sub-problem. Now, consider its KKT conditions:

$$\begin{cases}-\frac{H_{ij}}{B_{ij}}+\rho G_{ij}+\theta-\mu_j=0,\\ \sum_{j=1}^{n}B_{ij}=1,\\ B_{ij}\ge 0,\\ \mu_jB_{ij}=0,\\ \mu_j\ge 0.\end{cases} \tag{10}$$

Firstly, we compute θ by solving the equation $\sum_{j:H_{ij}\neq 0}\frac{H_{ij}}{\rho G_{ij}+\theta}=1$. Define $f(\theta)=\sum_{j:H_{ij}\neq 0}\frac{H_{ij}}{\rho G_{ij}+\theta}$. Since $H_{ii}>0$ and $G_{ii}=0$, $\lim_{\theta\to 0^+}f(\theta)\to+\infty$, and $\lim_{\theta\to+\infty}f(\theta)=0$. Moreover, $f(\theta)$ is a monotone decreasing function in the range $(0,+\infty)$. Thus $f(\theta)=1$ has and only has one solution in the range $(0,+\infty)$. We can find this solution by any root finding algorithms. After obtaining the solution θ, for any j such that $H_{ij}\neq 0$, we set $B_{ij}=\frac{H_{ij}}{\rho G_{ij}+\theta}\ge 0$ and $\mu_j=0$. For any other j, we set $B_{ij}=0$ and $\mu_j=\rho G_{ij}+\theta>0$. It is easy to verify that all KKT conditions are satisfied, and thus such B_{ij} is the global optima of Eq.(7).

When solving $\mathbf{E}$, we take $\mathbf{E}=\mathbf{B}-\mathbf{A}$ into Eq.(8) and find that it can also be de-coupled into n independent sub-problems. Consider the i-th one:

$$\begin{aligned}\min_{\mathbf{A_i}.} \quad & \sum_{j=1}^{n}\left(A_{ij}-\left(B_{ij}-\frac{\rho G_{ij}}{2\gamma}\right)\right)^2\\ s.t. \quad & 0\le A_{ij}\le 1,\quad \sum\nolimits_{j=1}^{n}A_{ij}=1. \end{aligned} \tag{11}$$

Note that Eq.(11) is a Euclidean projection onto a simplex, and thus its global optima can be obtained by a standard method as in (Nie, Wang, and Huang 2014). After obtaining $\mathbf{B}$ and $\mathbf{A}$, we can extract the noises $\mathbf{E}$ by $\mathbf{E}=\mathbf{B}-\mathbf{A}$ to achieve the graph level of robustness.

Optimizing F When optimizing $\mathbf{F}$, we need to solve the following sub-problem:

$$\begin{aligned} \min_{\mathbf{F}} \quad & tr(\mathbf{F}^T\mathbf{L}\mathbf{F}) \\ s.t. \quad & \mathbf{F}^T\mathbf{F} = \mathbf{I}. \end{aligned} \tag{12}$$

Eq.(12) can be solved by Ky Fan Theorem (Fan 1949), i.e., the closed-form solution is the c eigenvectors of $\mathbf{L}$ corresponding to the c smallest eigenvalues.

Optimizing $\boldsymbol{\alpha}$ When optimizing $\boldsymbol{\alpha}$, we obtain the following sub-problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{k=1}^{m} \alpha_k o_k, \\ s.t. \quad & \alpha_k \geq 0, \quad \sum_{k=1}^{m} \alpha_k^{-1} = 1. \end{aligned} \tag{13}$$

where $o_k = \sum_{i=1}^{n} w_i^2 \sum_{j=1}^{n} A_{ij}^{(k)} \log\left(\frac{A_{ij}^{(k)}}{A_{ij}+E_{ij}}\right)$. According to the Cauchy-Schwarz Inequality, we have

$$\sum_{k=1}^{m} \alpha_k o_k \sum_{k=1}^{m} \alpha_k^{-1} \geq \left(\sum_{k=1}^{m} \sqrt{o_k}\right)^2. \tag{14}$$

which leads to

$$\sum_{k=1}^{m} \alpha_k o_k \geq \left(\sum_{k=1}^{m} \sqrt{o_k}\right)^2. \tag{15}$$

The equality holds when $\alpha_k \propto \frac{1}{\sqrt{o_k}}$. So its closed form solution is:

$$\alpha_k = \frac{\sum_{j=1}^{m} \sqrt{o_j}}{\sqrt{o_k}}. \tag{16}$$

Note that o_k indicates the difference between the k-th base clustering result and the consensus one, which can be regarded as the quality of the k-th clustering result. The smaller it is, the better the k-th clustering result is. Since $\alpha_k \propto \frac{1}{\sqrt{o_k}}$, α_k indicates the quality of the k-th clustering result. If the k-th clustering result is better, then its weight α_k should be larger and the k-th clustering result plays a more important role in the ensemble learning. This achieves the base clustering level of robustness.

Algorithm

Algorithm 1 summarizes the whole TRCE algorithm. Although it involves some complicated sub-problems, we can find the global optima of each sub-problem. At last, we obtain the clustering result by directly finding the connective components on $\mathbf{A}$ without any uncertain post-processing like k-means or spectral clustering. Now, we analyse the time complexity of the proposed method. When optimizing $\mathbf{w}$, it costs $O(n^2m)$ time. When optimizing $\mathbf{B}$, we decompose it into n sub-problems, and in each sub-problem, we suppose the time complexity of root finding algorithm is $O(t)$. Then optimizing $\mathbf{B}$ costs $O(nt + n^2)$ time. The Euclidean projection used in optimizing $\mathbf{A}$ costs $O(n^2)$ time. Computing $\mathbf{F}$ costs $O(n^2c)$ time and computing $\boldsymbol{\alpha}$ costs $O(n^2m)$ time. Supposing the number of iterations is T, the whole time complexity is $O(T(n^2m + n^2c + nt))$, which is comparable with the existing graph based methods (Zhou et al. 2015b; Tao et al. 2017, 2019). Despite this, we plan to reduce the computation complexity in the future work.

Algorithm 1 TRCE Algorithm

Input: m base clustering results, number of clusters c, hyper-parameter λ.
Output: Consensus clustering result $\mathcal{C}$.

```
Construct the adjacent matrices S^(1), ..., S^(m) of the m multiple graphs from the base results.
Obtain m transition matrices A^(k) = (D^(k))^{-1} S^(k). Initialize A = (1/m) Σ_{k=1}^m A^(m), E = 0, α_k = m, γ = 1.
Construct Laplacian matrix L = D − (A + A^T)/2, and initialize F by solving Eq.(12)
while not converge do
    Compute w by Eq.(6). // Instance level of robustness
    Compute B by solving KKT conditions Eq.(10).
    Compute A by solving Eq.(11).
    Compute E by E = B − A. // Graph level of robustness
    Compute F by solving Eq.(12).
    Compute α by Eq.(16). // Base clustering level of robustness
    Update γ by γ = 1.1 * γ.
end while
Obtain the consensus clustering result C by finding the c connective components on A.
```

Datasets	#instances	#features	#classes
AR	840	768	120
Coil20	1440	1024	20
K1b	2340	21839	6
Lung	203	3312	5
Medical	706	1449	17
Tr41	878	7454	10
Tdt2	10212	36771	96
TOX	171	5748	4
UMIST	575	644	20
WarpAR	130	2400	10

Table 1: Information of the datasets.

Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed method.

Datasets

We use 10 datasets, including AR (Wang, Nie, and Huang 2014), Coil20 (Cai et al. 2010), K1b (Zhao and Karypis 2004), Lung (Hong and Yang 1991), Medical (Zhou et al. 2015b), Tr41 (Zhao and Karypis 2004), Tdt2 (Cai et al. 2007), TOX (Li et al. 2018), UMIST (Wechsler et al. 2012), WarpAR (Li et al. 2018). The detailed information of these datasets is shown in Table 1.

Experimental Setup

Following the experimental setup in (Zhou et al. 2015b), we run k-means 200 times with different random initialization-

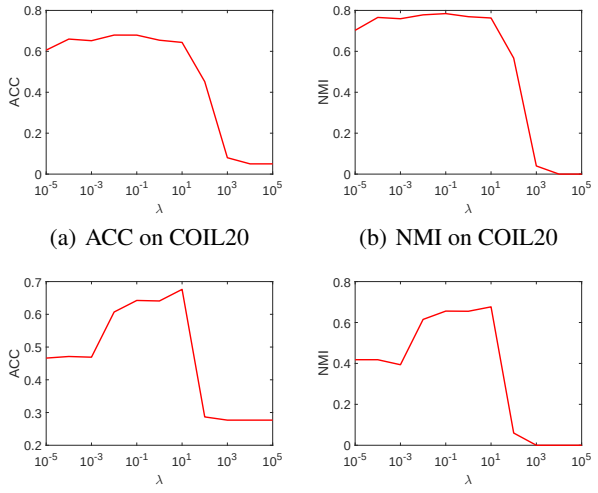
Methods	AR	COIL20	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WarpAR
KM	0.3026	0.5474	0.6726	0.7114	0.3996	0.5626	0.4104	0.4229	0.3974	0.2455
KM-best	0.3243	0.6350	0.8559	0.8675	0.4707	0.6946	0.4460	0.4825	0.4362	0.3192
CSPA	0.3439	0.6380	0.4531	0.4138	0.3500	0.5213	0.2850	0.4246	0.4068	0.2377
HGPA	0.3574	0.5501	0.5326	0.5025	0.2950	0.4894	0.2959	0.3854	0.4031	0.2523
MCLA	0.3427	0.6627	0.7383	0.7084	0.4017	0.5698	0.4000	0.4152	0.4050	0.2323
NMFC	0.3376	0.6464	0.5860	0.6764	0.3789	0.6323	0.3716	0.4269	0.3979	0.2231
RCE	0.3383	0.6295	0.6887	0.7143	0.3851	0.6391	-	0.4105	0.4059	0.2054
MEC	0.2782	0.5727	0.8190	0.7379	0.3627	0.6559	-	0.4304	0.4130	0.2592
LWEA	0.3236	0.5943	0.8279	0.7458	0.4208	0.6719	0.5744	0.4234	0.4052	0.2131
LWGP	0.3407	0.6547	0.7172	0.6498	0.4047	0.6483	0.4288	0.4193	0.4176	0.2292
RSEC	0.2990	0.5358	0.8409	0.8217	0.3490	0.6367	0.4222	0.4041	0.3127	0.2423
DREC	0.3394	0.5463	0.6462	0.6379	0.3926	0.6243	0.3684	0.4205	0.4174	0.2062
TRCE	**0.3619**	**0.7013**	**0.8899**	**0.9034**	**0.4356**	**0.6812**	**0.6273**	**0.4491**	**0.4410**	**0.2746**

Table 2: ACC results on all the datasets

s to obtain 200 base results. Then we divide them into 10 subsets, with 20 in each one. We apply clustering ensemble methods on each subset, and report the average results over the 10 subsets. We compare with the following methods:

- **KM**, which is the average result of all base clustering.
- **KM-best**, which is the best result of all base results.
- **CSPA** (Strehl and Ghosh 2003), which constructs a relationship between instances in the same cluster to establish a similarity measure for ensemble.
- **HGPA** (Strehl and Ghosh 2003), which combines base results with a constrained minimum cut objective.
- **MCLA**(Strehl and Ghosh 2003), which transforms the ensemble into a cluster correspondence problem.
- **NMFC** (Li and Ding 2008), which uses nonnegative matrix factorization to aggregate base results.
- **RCE** (Zhou et al. 2015b), which learns a robust consensus result via extracting the noises in the connective matrices.
- **MEC** (Tao et al. 2017), which learns a robust consensus result by using low-rank and sparse decomposition.
- **LWEA** (Huang, Wang, and Lai 2018), which is a hierarchical agglomerative clustering ensemble method based on local weighting strategy.
- **LWGP** (Huang, Wang, and Lai 2018), which is a graph partition clustering ensemble method based on the local weighting strategy.
- **RSEC** (Tao et al. 2019), which is a robust clustering ensemble method based on spectral clustering.
- **DREC** (Zhou, Zheng, and Pan 2019), which learns a dense representation for clustering ensemble.

On all data sets and for all methods, the number of clusters is set to the true number of classes. Our method adjusts γ automatically as introduced in Algorithm 1. The parameter ρ is also automatically decided. We first initialize $\rho = 1$, and then, if the rank of $\mathbf{L}$ is larger than $n - c$, which means the rank constraint is not strong enough, we double it. If its rank is smaller than $n - c$, i.e., the constraint is too strong, we reduce it by half. The only hyper-parameter tuned manually is λ. We tune it in $[10^{-5}, 10^{5}]$. We use Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate the clustering performance.



(a) ACC on COIL20 (b) NMI on COIL20

(c) ACC on Tr41 (d) NMI on Tr41

Figure 2: Clustering results on different values of λ.

Experimental Results

Tables 2 and 3 show the ACC and NMI results of all methods on all datasets, respectively. Note that, on the largest dataset Tdt2, RCE and MEC run out-of-memory error due to their high space complexity.

From Tables 2 and 3, we find that our TRCE can easily outperform KM, which shows the effectiveness of the ensemble. Moreover, on all datasets, TRCE also performs better than other state-of-the-art clustering ensemble methods, which demonstrates the superiority of our method.

Even compared with other robust clustering ensemble methods (RCE, MEC and RSEC), TRCE can achieve better performance. The reason may be in two folds. Firstly, the compared methods often pay attention on single level robust.

Methods	AR	COIL20	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WarpAR
KM	0.6560	0.7065	0.5493	0.5284	0.4209	0.5843	0.6111	0.1374	0.5855	0.2100
KM-best	0.6696	0.7527	0.6853	0.6558	0.4806	0.6713	0.6240	0.2164	0.6173	0.2960
CSPA	0.7030	0.7373	0.4071	0.3712	0.3992	0.5919	0.5589	0.1436	0.5806	0.2036
HGPA	0.7009	0.6710	0.3917	0.3372	0.3613	0.5084	0.5385	0.1083	0.5819	0.2183
MCLA	0.6914	0.7613	0.5944	0.5258	0.4296	0.6044	0.6070	0.1329	0.5962	0.1852
NMFC	0.6898	0.7597	0.4995	0.5202	0.4259	0.6512	0.5930	0.1434	0.5836	0.1968
RCE	0.6759	0.7588	0.6068	0.5248	0.4475	0.6499	-	0.1344	0.5951	0.1781
MEC	0.6098	0.7360	0.6818	0.5617	0.4089	0.6758	-	0.1313	0.5624	0.2058
LWEA	0.6712	0.7377	0.6948	0.5364	0.4185	0.6666	0.7183	0.1236	0.6055	0.1836
LWGP	0.6879	0.7639	0.6115	0.4993	0.4266	0.6535	0.6266	0.1333	0.6091	0.1951
RSEC	0.6030	0.7024	0.6615	0.6027	0.4036	0.6449	0.5243	0.1184	0.4214	0.1882
DREC	0.6770	0.7215	0.5774	0.4647	0.4510	0.6514	0.5971	0.1394	0.6040	0.1835
TRCE	**0.7412**	**0.7924**	**0.7496**	**0.7216**	**0.4622**	**0.6849**	**0.7275**	**0.1541**	**0.6500**	**0.3487**

Table 3: NMI results on all the datasets

Datasets	Metric	TRCE-α	TRCE-w	TRCE-E	TRCE
Lung	ACC	0.8936	0.8892	0.6847	0.9034
	NMI	0.7039	0.6972	0.0000	0.7216
Tdt2	ACC	0.6024	0.6023	0.1806	0.6273
	NMI	0.7102	0.7094	0.0000	0.7275

Table 4: Ablation study on Lung and Tdt2.

However, our method has a mechanism to control three levels of robustness: base clustering level, graph level and instance level. Therefore, TRCE provides a more sophisticated framework for robustness. Secondly, different from other robust methods which focus on contaminated data, the proposed one applies the self-paced learning which also considers the uncontaminated data. For those uncontaminated data, TRCE involves them in the ensemble learning in order of difficulty, so that the side effects caused by the difficult data would be alleviate because TRCE uses them for learning until it is strong enough. Therefore, our self-paced framework provides a more finely way to handle all instances (including the contaminated and uncontaminated ones).

At last, on most data sets, TRCE is comparable with or even better than KM-best. Note that, TRCE does not need to exhaustively search on the pre-defined pool of base clusterings. Moreover, it only takes base results as input without accessing original data or labels. This also well demonstrates the effectiveness of our method.

Ablation Study

As a tri-level method, we turn off each level respectively and get three degenerated versions: TRCE-α (without base clustering level, i.e., $\boldsymbol{\alpha} = \mathbf{1}$), TRCE-E (without graph level, i.e., $\mathbf{E} = \mathbf{0}$), and TRCE-w (without instance level, i.e., $\mathbf{w} = \mathbf{1}$). We show the results on Lung and Tdt2 in Table 4. The results on other datasets are similar.

From Table 4, we find that TRCE performs better than all three degenerated versions. It well demonstrates the effectiveness of the proposed tri-level model. Moreover, compared with TRCE-α and TRCE-w, TRCE-E gets poor results. As discussed before, if we drop $\mathbf{E}$, $\mathbf{A}$ will be very dense and may only contain one connective components, i.e., it puts all instances into one cluster. That is why the NMI of TRCE-E is often zero. So the graph level is the most crucial. By further considering the base clustering level and instance level robustness, the performance of the proposed method can be improved to some extent.

Parameter Study

In our method, there is only one hyper-parameter (λ) which needs to be tuned manually. Figure 2 shows the ACC and NMI results with different values of λ on COIL20 and Tr41 datasets. The results on other datasets are similar. From Figure 2, we find that our method can achieve good performance when λ is in the range $[10^{-2}, 10^{1}]$.

If λ is too small, $\mathbf{E}$ would be large, and thus the model may mistake useful information for noises. If λ is too large, to optimize the objective function Eq.(4), $\mathbf{E}$ should be close to all-zero matrix. As discussed before, the numbers of connective components may be smaller than c and caused poor performance. In the extreme case (e.g. $\lambda = 10^5$), there is only one connective component, which leads to the zero NMI.

Conclusion

In this paper, we proposed a novel tri-level robust clustering ensemble methods with self-paced multiple graph learning. We transformed clustering ensemble task to a multiple graph learning problem by constructing the transition matrices from the base clustering results. In the multiple graph learning, we considered three levels of robustness: base clustering level by assigning a weight on each base clustering result; graph level by directly extracting the noises on the consensus graph; instance level by applying self-paced learning on the data. We carefully designed a unified objective function to characterize the three levels of robustness simultaneously. Then we provided an effective algorithm to optimize the objective function. At last, we conducted extensive experiments on benchmark datasets and shew that the proposed TRCE outperformed other state-of-the-art clustering ensemble methods, including the robust clustering ensemble methods.

Acknowledgements

This work is supported by the National Natural Science Foundation of China grants 61806003, 61976205, 61976129, and 61972001; China National 973 program 2014CB340301; and Natural Science Foundation of Anhui Province grant 1908085MF188.

References

Bai, L.; Liang, J.; and Cao, F. 2020. A multiple *k*-means clustering ensemble algorithm to find nonlinearly separable clusters. *Inf. Fusion* 61: 36–47.

Cai, D.; He, X.; Han, J.; and Huang, T. S. 2010. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence* 33(8): 1548–1560.

Cai, D.; He, X.; Zhang, W. V.; and Han, J. 2007. Regularized Locality Preserving Indexing via Spectral Regression. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management (CIKM'07)*, 741–750.

Fan, K. 1949. On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations: II*. *Proceedings of the National Academy of Sciences of the United States of America* 36(1): 31–35.

Hong, Z.-Q.; and Yang, J.-Y. 1991. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *pattern recognition* 24(4): 317–324.

Huang, D.; Lai, J.; and Wang, C. 2016. Robust Ensemble Clustering Using Probability Trajectories. *IEEE Transactions on Knowledge and Data Engineering* 28(5): 1312–1326.

Huang, D.; Wang, C.; and Lai, J. 2018. Locally Weighted Ensemble Clustering. *IEEE Transactions on Systems, Man, and Cybernetics* 48(5): 1460–1473.

Jiang, L.; Meng, D.; Yu, S.; Lan, Z.; Shan, S.; and Hauptmann, A. G. 2014. Self-Paced Learning with Diversity 2078–2086.

Jiang, L.; Meng, D.; Zhao, Q.; Shan, S.; and Hauptmann, A. G. 2015. Self-paced curriculum learning. In *AAAI*, 2694–2700.

Kang, Z.; Pan, H.; Hoi, S. C. H.; and Xu, Z. 2020a. Robust Graph Learning From Noisy Data. *IEEE Trans. Cybern.* 50(5): 1833–1843.

Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J. T.; Peng, X.; Chen, W.; and Xu, Z. 2020b. Partition level multiview subspace clustering. *Neural Networks* 122: 279–288.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-Paced Learning for Latent Variable Models. In *NIPS*, 1189–1197.

Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Jing, L. 2019. Clustering ensemble based on sample's stability. *Artif. Intell.* 273: 37–55.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2018. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50(6): 94.

Li, T.; and Ding, C. H. Q. 2008. Weighted Consensus Clustering. In *SDM*, 798–809.

Li, T.; Ding, C. H. Q.; and Jordan, M. I. 2007. Solving Consensus and Semi-supervised Clustering Problems Using Nonnegative Matrix Factorization. In *ICDM*, 577–582.

Liu, H.; Liu, T.; Wu, J.; Tao, D.; and Fu, Y. 2015. Spectral Ensemble Clustering. In *SIGKDD*, 715–724.

Liu, H.; Shao, M.; Li, S.; and Fu, Y. 2018. Infinite ensemble clustering. *Data Mining and Knowledge Discovery* 32(2): 385–416.

Meng, D.; Zhao, Q.; and Jiang, L. 2017. A theoretical understanding of self-paced learning. *Information Sciences* 414: 319–328.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*, 977–986.

Ren, Y.; Que, X.; Yao, D.; and Xu, Z. 2019. Self-paced multi-task clustering. *Neurocomputing* 350: 212–220.

Strehl, A.; and Ghosh, J. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3(3): 583–617.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From Ensemble Clustering to Multi-View Clustering. In *IJCAI*, 2843–2849.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2019. Robust Spectral Ensemble Clustering via Rank Minimization. *ACM Transactions on Knowledge Discovery From Data* 13(1): 1–25.

Wang, H.; Nie, F.; and Huang, H. 2014. Globally and locally consistent unsupervised projection. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; and Yin, J. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 3778–3784. ijcai.org.

Wechsler, H.; Phillips, J. P.; Bruce, V.; Soulié, F. F.; and Huang, T. S. 2012. *Face recognition: From theory to applications*, volume 163. Springer Science & Business Media.

Yu, Z.; Chen, H.; You, J.; Liu, J.; Wong, H.; Han, G.; and Li, L. 2015. Adaptive Fuzzy Consensus Clustering Framework for Clustering Analysis of Cancer Data. *IEEE/ACM Trans. Comput. Biology Bioinform.* 12(4): 887–901.

Zhang, D.; Meng, D.; and Han, J. 2017. Co-Saliency Detection via a Self-Paced Multiple-Instance Learning Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(5): 865–878.

Zhao, Y.; and Karypis, G. 2004. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning* 55(3): 311–331.

Zhou, J.; Zheng, H.; and Pan, L. 2019. Ensemble clustering based on dense representation. *Neurocomputing* ISSN 0925-2312.

Zhou, P.; Du, L.; and Li, X. 2020. Self-paced Consensus Clustering with Bipartite Graph. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2133–2139. ijcai.org.

Zhou, P.; Du, L.; Liu, X.; Shen, Y. D.; Fan, M.; and Li, X. 2020. Self-Paced Clustering Ensemble. *IEEE Transactions on Neural Networks and Learning Systems* 1–15. doi:10.1109/TNNLS.2020.2984814.

Zhou, P.; Du, L.; Shi, L.; Wang, H.; and Shen, Y.-D. 2015a. Recovery of corrupted multiple kernels for clustering. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Zhou, P.; Du, L.; Wang, H.; Shi, L.; and Shen, Y. 2015b. Learning a robust consensus matrix for clustering ensemble via Kullback-Leibler divergence minimization. In *IJCAI*, 4112–4118.