A Non-parametric Graph Clustering Framework for Multi-View Data

Shengju Yu[1], Siwei Wang[2], Zhibin Dong[1], Wenxuan Tu[1], Suyuan Liu[1], Zhao Lv[2*], Pan Li[2], Miao Wang[2*], En Zhu[1*]

[1]School of Computer, National University of Defense Technology, Changsha, 410073, China
[2]Intelligent Game and Decision Lab, Beijing, 100071, China
yu-shengju@foxmail.com, {zhaolv, mercury.miao, enzhu}@nudt.edu.cn

Abstract

Multi-view graph clustering (MVGC) derives encouraging grouping results by seamlessly integrating abundant information inside heterogeneous data, and has captured surging focus recently. Nevertheless, the majority of current MVGC works involve at least one hyper-parameter, which not only requires additional efforts for tuning, but also leads to a complicated solving procedure, largely harming the flexibility and scalability of corresponding algorithms. To this end, in the article we are devoted to getting rid of hyper-parameters, and devise a non-parametric graph clustering (NpGC) framework to more practically partition multi-view data. To be specific, we hold that hyper-parameters play a role in balancing error item and regularization item so as to form high-quality clustering representations. Therefore, under without the assistance of hyper-parameters, how to acquire high-quality representations becomes the key. Inspired by this, we adopt two types of anchors, view-related and view-unrelated, to concurrently mine exclusive characteristics and common characteristics among views. Then, all anchors' information is gathered together via a consensus bipartite graph. By such ways, NpGC extracts both complementary and consistent multi-view features, thereby obtaining superior clustering results. Also, linear complexities enable it to handle datasets with over 120000 samples. Numerous experiments reveal NpGC's strong points compared to lots of classical approaches.

Introduction

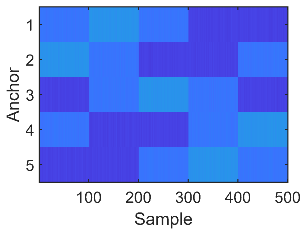
Along with the advance of information age, multi-view data which generally comes from heterogeneous modalities or various channels of the same instances is growingly ubiquitous (Yang et al. 2021b; Wan et al. 2022; Liang et al. 2023b; Fu et al. 2023; Yu et al. 2023b; Wang et al. 2022a; Liang et al. 2023a; Wan et al. 2023; Fu et al. 2022; Xia et al. 2023). Correspondingly, how to comprehensively exploit the information hidden into these data is becoming a research highlight. Multi-view clustering (MVC), a representative unsupervised learning technique, is deemed as a strong instrument to effectively mine the intrinsic structure and is popularly employed in drug discovery, protein prediction, medical image analysis, etc (Xia et al. 2021b; Yu et al. 2023a; Wen et al. 2022). Co-training, kernel, matrix factorization, graph and neural network are five common types of solutions for MVC problems. Co-training approaches (Nie, Shi, and Li 2020; Kumar, Rai, and Daume 2011; Huang et al. 2020; Du et al. 2021) alternatively exchange the clustering results on each view and try to reach agreement between views. Kernel approaches (Li et al. 2022a; Kang et al. 2018; Liu et al. 2021; Wang et al. 2021) jointly optimize a set of preset kernels and try to generate a consistent optimal kernel. Matrix factorization approaches (Zhang et al. 2021; Huang, Kang, and Xu 2020; Luong et al. 2022; Yang et al. 2020) decrease the data dimension and try to seek the unified potential features in low-dimensional space. Graph approaches (Khan and Maji 2021; Wang, Yang, and Liu 2020; Liu et al. 2022; Yang et al. 2022a; Shi et al. 2021) make use of graph structure to characterize the pair-wise affinities and try to construct the unified similarity graph for all views. Neural network approaches (Qin et al. 2021; Xia et al. 2021a; Tu et al. 2021; Yang et al. 2022b; Wen et al. 2023) capture advanced features by ingenious architectures and try to learn the common high-level characteristics.

Despite appreciable results, the majority of them suffer from at least one hyper-parameter, which not only brings extra tuning overheads but also results in overly-complex solution process, heavily limiting their further deployment in numerous scenarios. Besides, how to set the appropriate value range is also an intractable problem. Consequently, designing non-parametric methods becomes an urgent need. To meet this need, in the paper we first investigate the role of hyper-parameters. Taking the classical MVGC framework for example,

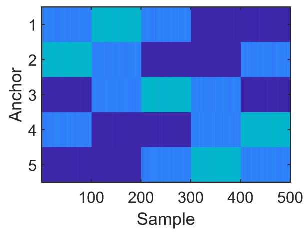
$$\begin{aligned}\min_{\mathbf{Q}^{(r)},\mathbf{Q}} &\left\|\mathbf{X}^{(r)}-\mathbf{B}^{(r)}\mathbf{Q}^{(r)}\right\|_F^2+\lambda\left\|\mathbf{Q}-\mathbf{Q}^{(r)}\right\|_F^2\\ s.t.\ &\left(\mathbf{Q}^{(r)}\right)^{\top}\mathbf{1}=\mathbf{1},\mathbf{Q}^{(r)}\geq 0,\mathbf{Q}^{\top}\mathbf{1}=\mathbf{1},\mathbf{Q}\geq 0,\end{aligned}\tag{1}$$

where $\mathbf{X}^{(r)}$, $\mathbf{B}^{(r)}$, $\mathbf{Q}^{(r)}$ and $\mathbf{Q}$ represent the data, anchor matrix, bipartite graph on view r and the merged bipartite graph respectively, we know that λ aims at balancing $\left\|\mathbf{X}^{(r)}-\mathbf{B}^{(r)}\mathbf{Q}^{(r)}\right\|_F^2$ and $\left\|\mathbf{Q}-\mathbf{Q}^{(r)}\right\|_F^2$ so that the generated $\mathbf{Q}$ is favorable for clustering. Especially, we present $\mathbf{Q}$ and its complete graph learned on a synthetic dataset that is with 500 samples, 2 views and 5 clusters, as shown in Figure 1. As seen, λ can influence the sample-anchor similarity structure of the merged bipartite graph $\mathbf{Q}$, and thereby

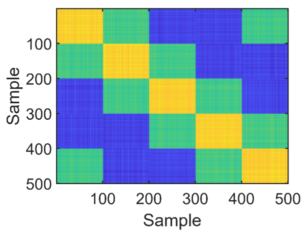
*Corresponding authors



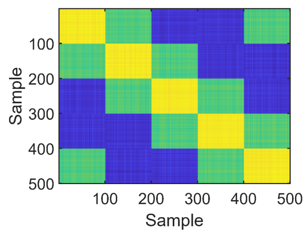
(a) $\lambda = 10$, $\mathbf{Q}$ (ACC=72.24%)



(b) $\lambda = 10^3$, $\mathbf{Q}$ (ACC=85.28%)



(c) $\lambda = 10$, $\mathbf{C}$ (ACC=72.24%)



(d) $\lambda = 10^3$, $\mathbf{C}$ (ACC=85.28%)

Figure 1: The role of hyper-parameter λ. Different λ values bring different graph structures and clustering results. Complete graph $\mathbf{C}$ of the graph $\mathbf{Q}$ is restored by setting $\mathbf{C} = \mathbf{Q}^\top \Delta^{-1} \mathbf{Q}$ where the diagonal matrix Δ is with $\Delta_{i,i} = \sum_{j=1}^{n} \mathbf{Q}_{i,j}$. Clearer sample-anchor structure of $\mathbf{Q}$ or block diagonal structure of $\mathbf{C}$ indicates preferable results.

brings different clustering results. Moreover, $\mathbf{Q}$ with more distinct structure (equivalently, its complete graph $\mathbf{C}$ has clearer block diagonal structure) generates more preferable results. Therefore, how to acquire high-quality $\mathbf{Q}$ is the key to design pleasing non-parametric clustering algorithms.

Inspired by the above analysis, we design a NpGC framework to form desirable $\mathbf{Q}$, as shown in Figure 2. To be specific, in addition to the view-related anchor matrices $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$, we introduce a view-unrelated anchor matrix $\mathbf{B}$. $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ and $\mathbf{B}$ are utilized to concurrently mine view-exclusive characteristics and view-common characteristics. $\{\mathbf{S}^{(r)}\}_{r=1}^{v}$ are a group of projections, which aim at finding the common representation space for all views. Furthermore, orthogonal to constructing $\mathbf{Q}$ by merging all $\{\mathbf{Q}^{(r)}\}_{r=1}^{v}$, we directly construct $\mathbf{Q}$ based on $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ and $\mathbf{B}$. One positive is that this can decrease the computational expenditure by avoiding the generation of $\{\mathbf{Q}^{(r)}\}_{r=1}^{v}$. Another is that this relieves the information loss caused by the mergence operation. Most importantly, $\mathbf{Q}$ extracts heterogeneous representations directly at anchor level rather than at bipartite graph level, which can maintain original diversity information and improve $\mathbf{Q}$'s richness. Also, it acts as a bridge for $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ and $\mathbf{B}$ to make them able to mutually negotiate such that the produced anchors are more distinctive. By such ways, $\mathbf{Q}$ contains both complementary and consistent multi-view features, thereby achieving superior clustering results. Owing to not involving the mergence operation in our framework, we rename $\mathbf{Q}$ consensus bipartite graph. Beyond these, NpGC is demonstrated to have linear complexities, which enables it to efficiently tackle the datasets with

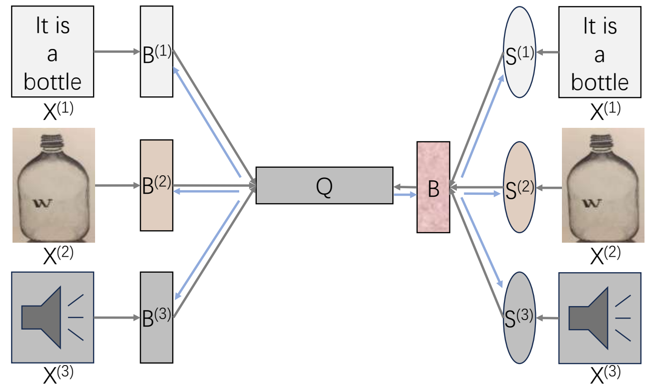


Figure 2: Our NpGC framework. $\{\mathbf{X}^{(r)}\}_{r=1}^{3}$: multi-view data; $\{\mathbf{B}^{(r)}\}_{r=1}^{3}$: view-related anchor matrices; $\mathbf{B}$: view-unrelated anchor matrix; $\{\mathbf{S}^{(r)}\}_{r=1}^{3}$: view-related projections; $\mathbf{Q}$: consensus bipartite graph. $\mathbf{Q}$ is directly constructed via $\{\mathbf{B}^{(r)}\}_{r=1}^{3}$ and $\mathbf{B}$, thus gathering exclusive characteristics and common characteristics among views. $\{\mathbf{B}^{(r)}\}_{r=1}^{3}$, $\mathbf{B}$, $\mathbf{Q}$ and $\{\mathbf{S}^{(r)}\}_{r=1}^{3}$ are learned in a mutual enhancement way to progress towards higher-quality outputs.

over 120000 instances. Interestingly, when there is only one view data, NpGC still works properly by simply setting v as 1. Therefore, it can be extended to single-view clustering problems. A series of experiments give evidence of NpGC's strong points against 17 prominent approaches.

Proposed Model and Solution

Our NpGC is formulated as

$$\min_{\mathbf{B}^{(r)},\mathbf{B},\mathbf{S}^{(r)},\mathbf{Q}} \sum_{r=1}^{v} \left\|\mathbf{X}^{(r)} - \mathbf{B}^{(r)}\mathbf{Q}\right\|_F^2 + \left\|\mathbf{S}^{(r)}\mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q}\right\|_F^2$$
$$s.t.\ \mathbf{Q}^\top \mathbf{1} = \mathbf{1}, \mathbf{Q} \geq 0, \mathbf{S}^{(r)} \left(\mathbf{S}^{(r)}\right)^\top = \mathbf{I}_k, \tag{2}$$

where $\mathbf{X}^{(r)} \in \mathbb{R}^{d_r \times n}$, $\mathbf{B}^{(r)} \in \mathbb{R}^{d_r \times k}$, $\mathbf{Q} \in \mathbb{R}^{k \times n}$, $\mathbf{S}^{(r)} \in \mathbb{R}^{k \times d_r}$ and $\mathbf{B} \in \mathbb{R}^{k \times k}$. d_r, n and k are the data dimension on view r, the sample number and the cluster number.

Solution

Due to the objective (2) being non-convex, we generate the solutions by splitting it into different sub-problems:

$\mathbf{B}^{(r)}$ Sub-problem Keeping $\mathbf{B}$, $\mathbf{S}^{(r)}$ and $\mathbf{Q}$ constant, minimizing the objective (2) is equivalent to solving

$$\min_{\mathbf{B}^{(r)}} \left\|\mathbf{X}^{(r)} - \mathbf{B}^{(r)}\mathbf{Q}\right\|_F^2. \tag{3}$$

This is an optimization problem without any constraints, and the optimal solution can be acquired by setting its derivative to zero. That is,

$$\left(\mathbf{X}^{(r)} - \mathbf{B}^{(r)}\mathbf{Q}\right)\mathbf{Q}^\top = \mathbf{0}. \tag{4}$$

Since the rank of bipartite graph $\mathbf{Q}$ is k, $\mathbf{Q}\mathbf{Q}^\top \in \mathbb{R}^{k \times k}$ is reversible. Therefore, we have

$$\mathbf{B}^{(r)} = \mathbf{X}^{(r)}\mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top\right)^{-1}. \tag{5}$$

B Sub-problem For $\mathbf{B}$, we need to optimize

$$\min_{\mathbf{B}} \sum_{r=1}^{v} \left\| \mathbf{S}^{(r)} \mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q} \right\|_F^2 . \tag{6}$$

It also is an optimization problem without any constraints, and thus the optimal solution is

$$\mathbf{B} = \frac{1}{v} \left(\sum_{r=1}^{v} \mathbf{S}^{(r)} \mathbf{X}^{(r)} \right) \mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top \right)^{-1} . \tag{7}$$

$\mathbf{S}^{(r)}$ Sub-problem During updating $\mathbf{S}_r$, the objective (2) is equivalently formed as

$$\min_{\mathbf{S}^{(r)}} \left\| \mathbf{S}^{(r)} \mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q} \right\|_F^2 \;\; s.t. \;\; \mathbf{S}^{(r)} \left(\mathbf{S}^{(r)} \right)^\top = \mathbf{I}_k , \tag{8}$$

which is an optimization problem with the orthogonal constraint.

For solving it, we first introduce the column-orthogonal matrix $\mathbf{H}_r \in \mathbb{R}^{d_r \times k}$, and the problem (8) is equivalent to

$$\min_{\mathbf{H}^{(r)}} \left\| \left(\mathbf{H}^{(r)} \right)^\top \mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q} \right\|_F^2 \;\; s.t. \;\; \left(\mathbf{H}^{(r)} \right)^\top \mathbf{H}^{(r)} = \mathbf{I}_k . \tag{9}$$

Afterwards, we have

$$\begin{aligned} \left\| \left(\mathbf{H}^{(r)} \right)^\top \mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q} \right\|_F^2 &\le \left\| \left(\mathbf{H}^{(r)} \right)^\top \right\|_F^2 \left\| \mathbf{X}^{(r)} - \mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \right\|_F^2 \\ &= k \left\| \mathbf{X}^{(r)} - \mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \right\|_F^2 . \end{aligned} \tag{10}$$

Therefore, we can solve the problem (8) by minimizing its upper bound. Furthermore, we have

$$\begin{aligned} &\min_{\mathbf{H}^{(r)}} \left\| \mathbf{X}^{(r)} - \mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \right\|_F^2 = T_{RACE} \left(\left(\mathbf{X}^{(r)} \right)^\top \mathbf{X}^{(r)} \right. \\ &\left. -2 \left(\mathbf{X}^{(r)} \right)^\top \mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} + \mathbf{Q}^\top \mathbf{B}^\top \left(\mathbf{H}^{(r)} \right)^\top \mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \right) \\ &\Leftrightarrow \max_{\mathbf{H}^{(r)}} T_{RACE} \left(\mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \left(\mathbf{X}^{(r)} \right)^\top \right) , \end{aligned} \tag{11}$$

where $T_{RACE}(\cdot)$ is the trance operation.

Let the single value decomposition of $\mathbf{B}\mathbf{Q} \left(\mathbf{X}^{(r)} \right)^\top$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, and we obtain

$$T_{RACE} \left(\mathbf{H}^{(r)} \mathbf{B}\mathbf{Q} \left(\mathbf{X}^{(r)} \right)^\top \right) = T_{RACE} \left(\mathbf{\Sigma}\mathbf{D} \right) , \tag{12}$$

where $\mathbf{D} = \mathbf{V}^\top \mathbf{H}^{(r)} \mathbf{U}$. Additionally, given $\mathbf{D}^\top \mathbf{D} = \mathbf{U}^\top \left(\mathbf{H}^{(r)} \right)^\top \mathbf{V}\mathbf{V}^\top \mathbf{H}^{(r)} \mathbf{U} = \mathbf{I}$ and the non-negativeness of $\mathbf{\Sigma}$, we have

$$T_{RACE} \left(\mathbf{\Sigma}\mathbf{D} \right) \le T_{RACE} \left(\mathbf{\Sigma} \right) , \tag{13}$$

where the equality holds when the diagonal elements of $\mathbf{D}$ take 1. So, the optimal $\mathbf{H}^{(r)}$ is the product of $\mathbf{V}$ and $\mathbf{U}^\top$. $\mathbf{S}^{(r)}$ is obtained by setting it as the transport of $\mathbf{H}^{(r)}$.

Algorithm 1: Solution to the problem (2)

Input: Multi-view data $\{\mathbf{X}^{(r)}\}_{r=1}^{v}$.
Output: $\mathbf{Q}$.
Initialize: $\mathbf{B}^{(r)}, \mathbf{B}, \mathbf{S}^{(r)}, \mathbf{Q}$.
1: Let $t = 1$, $f_{obj}(0) = 10^{20}$.
2: **while** $\left[(f_{obj}(t) - f_{obj}(t-1))/f_{obj}(t) \le 10^{-4} \right]$ **do**
3: Update $\mathbf{B}^{(r)}$ via (5).
4: Update $\mathbf{B}$ via (7).
5: Update $\mathbf{S}^{(r)}$ via (8).
6: Update $\mathbf{Q}$ via (16).
7: $t = t + 1$.
8: **end while**

Q Sub-problem When solving $\mathbf{Q}$, it is equivalent to

$$\begin{aligned} \min_{\mathbf{Q}} \sum_{r=1}^{v} \left\| \mathbf{X}^{(r)} - \mathbf{B}^{(r)} \mathbf{Q} \right\|_F^2 + \left\| \mathbf{S}^{(r)} \mathbf{X}^{(r)} - \mathbf{B}\mathbf{Q} \right\|_F^2 \\ s.t. \;\; \mathbf{Q}^\top \mathbf{1} = \mathbf{1}, \mathbf{Q} \ge 0. \end{aligned} \tag{14}$$

After unfolding F-norm terms, the problem (14) equals to

$$\begin{aligned} &\min_{\mathbf{Q}} T_{RACE} \left(\mathbf{Q}^\top \left(\sum_{r=1}^{v} \left(\mathbf{B}^{(r)} \right)^\top \mathbf{B}^{(r)} + \sum_{r=1}^{v} \mathbf{B}^\top \mathbf{B} \right) \mathbf{Q} \right. \\ &\left. -2 \left(\sum_{r=1}^{v} \left(\mathbf{B}^{(r)} \right)^\top \mathbf{X}^{(r)} + \sum_{r=1}^{v} \mathbf{B}^\top \mathbf{S}^{(r)} \mathbf{X}^{(r)} \right) \mathbf{Q}^\top \right) \\ &\qquad s.t. \;\; \mathbf{Q}^\top \mathbf{1} = \mathbf{1}, \mathbf{Q} \ge 0. \end{aligned} \tag{15}$$

It can be split into n sub-problems according to the feasible region:

$$\begin{aligned} &\min_{\mathbf{Q}_{:,j}} \frac{1}{2} \mathbf{Q}_{:,j}^\top \left(\sum_{r=1}^{v} \left(\mathbf{B}^{(r)} \right)^\top \mathbf{B}^{(r)} + \sum_{r=1}^{v} \mathbf{B}^\top \mathbf{B} \right) \mathbf{Q}_{:,j} \\ &- \left(\left(\sum_{r=1}^{v} \left(\mathbf{B}^{(r)} \right)^\top \mathbf{X}^{(r)} + \sum_{r=1}^{v} \mathbf{B}^\top \mathbf{S}^{(r)} \mathbf{X}^{(r)} \right)_{:,j} \right)^\top \mathbf{Q}_{:,j} \\ &\qquad s.t. \;\; \mathbf{Q}_{:,j}^\top \mathbf{1} = 1, \mathbf{Q}_{:,j} \ge 0, j = 1, 2, 3, \ldots, n, \end{aligned} \tag{16}$$

which is a quadratic programming (QP) task and can be solved in $\mathcal{O}(k^3)$ time cost.

The whole solution to the problem (2) is summarized in Algorithm 1. $f_{obj}(t)$ is the objective value at t-th iteration.

Algorithm Analysis

Convergence: Algorithm 1 decomposes the problem (2) into four sub-problems, and each sub-problem can obtain its optimal solution. Therefore, the objective value is monotonically non-ascending during solving each sub-problem. Besides, the objective (2) has the lower bound, like 0. Given the alternating optimization theory (Bezdek and Hathaway 2003), Algorithm 1 is convergent.

Space Complexity: During the optimization process for the problem (2), storing matrices $\mathbf{B}^{(r)}$, $\mathbf{B}$, $\mathbf{S}^{(r)}$ and $\mathbf{Q}$, $r =$

$1, 2, \cdots, n$, consumes $\mathcal{O}(dk)$, $\mathcal{O}(k^2)$, $\mathcal{O}(kd)$ and $\mathcal{O}(kn)$ spaces respectively. d and k are constants, and generally are smaller than n. Therefore, the space complexity of Algorithm 1 is $\mathcal{O}(kn)$. Apparently, it is linear to n.

Time Complexity: It consumes $\mathcal{O}(k^2n + k^3)$ time expenditure to construct $\mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top\right)^{-1}$. Accordingly, constructing $\mathbf{X}^{(r)}\mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top\right)^{-1}$ costs $\mathcal{O}(d_rnk + k^2n + k^3)$. Therefore, solving the sub-problem $\mathbf{B}_r$ consumes $\mathcal{O}(dnk + k^2n + k^3)$ expenditure. Then, it costs $\mathcal{O}(kdn)$ complexity to compute $\sum_{r=1}^{v} \mathbf{S}^{(r)}\mathbf{X}^{(r)}$. The multiplication between $\sum_{r=1}^{v} \mathbf{S}^{(r)}\mathbf{X}^{(r)}$ and $\mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top\right)^{-1}$ costs $\mathcal{O}(k^2n)$. Since $\mathbf{Q}^\top \left(\mathbf{Q}\mathbf{Q}^\top\right)^{-1}$ has been constructed when solving $\mathbf{B}^{(r)}$, it will cost $\mathcal{O}(kdn + k^2n)$ expenditure to obtain the optimal $\mathbf{B}$. Subsequently, constructing $\mathbf{X}^{(r)}\mathbf{Q}^\top\mathbf{B}^\top$ and performing singular value decomposition on it cost $\mathcal{O}(d_rnk + nk^2)$ and $\mathcal{O}(d_rk^2)$ respectively. Since k is largely smaller than n, generating the optimal $\mathbf{S}^{(r)}$ costs $\mathcal{O}(d_rnk + nk^2)$. Thus, solving the sub-problem $\mathbf{S}^{(r)}$ costs $\mathcal{O}(dnk + nk^2)$ expenditure. Afterwards, constructing $\sum_{r=1}^{v} \left(\mathbf{B}^{(r)}\right)^\top \mathbf{B}^{(r)}$, $\mathbf{B}^\top\mathbf{B}$, $\sum_{r=1}^{v} \left(\mathbf{B}^{(r)}\right)^\top \mathbf{X}^{(r)}$ and $\mathbf{B}^\top \sum_{r=1}^{v} \mathbf{S}^{(r)}\mathbf{X}^{(r)}$ cost $\mathcal{O}(k^2d)$, $\mathcal{O}(k^3)$, $\mathcal{O}(kdn)$ and $\mathcal{O}(k^2n + kdn)$ respectively. Each column of $\mathbf{Q}$ is optimized by solving a QP problem, which costs $\mathcal{O}(k^3)$ time expenditure. So, solving the sub-problem $\mathbf{Q}$ costs $\mathcal{O}(k^2d+k^3+kdn+k^2n+nk^3)$. Based on the above analysis, we obtain that Algorithm 1 has $\mathcal{O}(dnk+nk^3)$ time complexity, which is linear to n.

Compared Methods and Datasets

Compared Methods

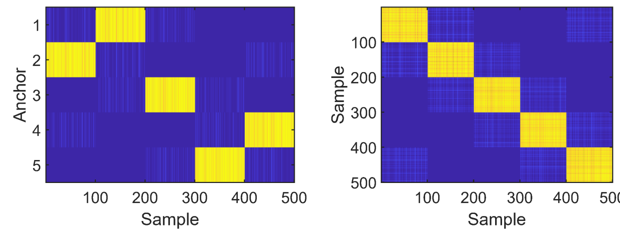
For sufficiently showing NpGC's strong points, the following 17 classical algorithms are selected for comparison:

1. **MVSC (Gao et al. 2015).** It concurrently clusters the subspace representation of each view, and ensures the instances in different views to be grouped into the same cluster by using the common indicator.
2. **AMGL (Nie, Li, and Li 2016).** It automatically learns a group of weights for all graphs so as to maintain the diversity, and guarantees the objective to be convex by reformulating the standard spectral clustering process.
3. **MLRSSC (Brbić and Kopriva 2018).** It learns the joint subspace representation by building up a single similarity matrix, and takes advantages of the sparsity and low-rank constraints to alleviate the noise affect.
4. **MSCIAS (Wang et al. 2019).** It establishes an intactness-aware affinity in the intact space to alleviate the information loss of unbalanced views, and adopts HSIC criterion instead of spectral clustering to increase its discrimination.
5. **BMVC (Zhang et al. 2019).** It alternatively learns the binary cluster structures and collaborative codes of multi-view data, and improves the efficiency by employing the code balance constraints on the cluster center.
6. **FMR (Li et al. 2019).** It maps the sample features into kernel space to exploit the non-linear dependence relationships, and generates the potential representation via the kernel correlation measure.
7. **mPAC (Kang et al. 2019).** It leverages multi-view representations in partition space rather than in data space, and aligns every partition through the rotation matrix to construct a unified indicator matrix.
8. **MCLES (Chen et al. 2020).** It clusters data in a learnable potential embedding space rather than in the original space, and jointly learns the global structure and indicator matrix to integrate the complementary information.
9. **PMSC (Kang et al. 2020).** It makes use of multiview data information in partition level instead of at graph generation stage to highlight the representation ability, and directly combines partitions into one to explore the cluster structure.
10. **PFSC (Lv et al. 2021).** It fuses multiple view-partitions as a substitute for the single graph to relieve the inconsistencies between heterogeneous representations, and utilizes the intrinsic interactions among subtasks to enhance the individual partition's discrimination.
11. **FMCNOF (Yang et al. 2021a).** It combines graph method and matrix factorization strategy together to boost the efficiency, and skips the post-processing stage of cluster label assignment by directly factoring the indicator matrix.
12. **FPMVS (Wang et al. 2022b).** It explores the consistent potential data distribution by mapping consensus landmarks into original data space, and introduces view-related coefficient to adaptively measure the importance between views.
13. **SFMC (Li et al. 2022b).** It fuses view-wise graphs in a self-supervised weighting manner rather than by regularizing them, and directly outputs the clusters by imposing a connectivity constraint on the joint graph.
14. **MSGL (Kang et al. 2022).** It introduces a view-specific weighting scheme to highlight the contributions of different views, and explicitly exploits the cluster structure of learned graph through a Laplacian matrix constraint.
15. **PGSC (Wu et al. 2023).** It constructs pure graphs by searching the optimal neighbors to achieve the connectivity and sparsity concurrently, and unifies the graph generation and label allocation to dispel the adverse impact of separated graphs on the clustering results.
16. **UDBGL (Fang et al. 2023).** It boosts the distinction of learned representations by jointly solving single-view and consensus-view graphs, and breaks down the k-means step's impact by directly producing the discrete clusters using one rank constraint.
17. **EEOMVC (Wang et al. 2023).** It mitigates the noise and redundancy by aggregating potential information at graph partition stage instead of at similarity stage, and decreases the risk of subsequent discrete procedure in virtue of the binary indicators.

Multi-view Datasets

All experiments are conducted on the following popular datasets, with sample sizes ranging from 512 to 126054:

1. Calte101view3 [1]: This dataset contains 512 image objects, including elephant, camera, aircraft and etc. It is with 11 classes and 3 views, and the data dimensions are 254, 512 and 36 respectively.
2. CCV[2]: This video dataset is collected under diverse illumination and imaging circumstances, and is composed of 6773 samples from 20 categories. The data dimensions on 3 views are all 20.
3. STL10[3]: This is a object recognition dataset including car, horse, ship and etc, and consists of 13000 samples from 10 different categories. The data dimensions on 3 views are 1024, 512 and 2048 respectively.
4. VGGFace100[4]: This dataset consists of 36287 faces with various profession, illumination, pose, ethnicity and age. The number of classes is 100, and the data dimensions on 4 views are 944, 576, 512 and 640 respectively.
5. YTF20[5]: These are 63896 heterogeneous face identification samples extracted from YouTube, and they are with 20 clusters. The data dimensions on 4 views are 944, 576, 512 and 640 respectively.
6. YTF50[6]: There are 126054 face video samples from 50 different categories. Their data dimensions on 4 views are 944, 576, 512 and 640 respectively.



(a) **Q** (ACC=93.02%) (b) **C** (ACC=93.02%)

Figure 3: The learned graph structure on synthetic dataset.

Experiments

Results and Analysis

We first display the graph structure learned on the synthetic dataset. As seen in Figure 3, NpGC learns clearer sample-anchor structure and block diagonal structure than those in Figure 1, which illustrates that it can capture higher-quality clustering representations. Then, we report the clustering results on 6 real datasets in Table 1 and 2. Several observations could be drawn from these tables:

1. **NpGC** in most situations is able to generate the most satisfactory results in comparison with existing remarkable methods. For instance, it achieves the improvements of 0.64%, 0.23%, 4.66%, 1.18%, 2.03% and 2.05% in terms of Precision than the Rank-2 counterparts like **PFSC**, **mPAC**, **UDBGL**, **EEOMVC** and **BMVC** respectively.
2. **NpGC** is with smaller standard deviation compared to several stronger competitors like **PFSC**, **FPMVS** and **MSCIAS**, which indicates that **NpGC**'s performance is relatively stable. Methods **mPAC**, **FMCNOF**, **SFMC**, **MSGL** and so on have 0 deviation value. This is mainly because they directly output the clustering label rather than performing post-processing operations on the representations. Despite very stable, these methods typically have sub-optimal results than ours.
3. **NpGC** receives a few second-best results on STL10, YTF20 and YTF50 against **BMVC** and **UDBGL**, the reasons of which could be that **BMVC** and **UDBGL** introduce some view adjustment policies to balance the contributions or importance of different views.
4. **NpGC** can consistently generate worth-having results on these datasets with sample size from 512 to 126054, which gives evidence that NpGC is resource-economical and practical in handling various MVC problems.
5. Methods **MVSC**, **AMGL**, **MLRSSC**, **MSCIAS**, **SFMC**, **PGSC**, **EEOMVC** and so on can not properly run on YTF20 and YTF50, which suggests that they encounter limited applicability and are incapable of coping with the MVC problems with large scale samples. Orthogonal to them, **NpGC** can get rid of this dilemma.
6. Methods **FMCNOF**, **FPMVS** and **MSGL** properly run on all datasets, and yet usually output the inferior results compared with **NpGC**. The reasons are possibly that they aggregate multi-view representations by merging graphs generated by single-level anchors. Although consuming fewer storage space, this results in the diversity information being weakened.
7. Methods **AMGL**, **FPMVS** and **SFMC** also do not involve hyper-parameters, and however all of them provide poorer results with significant margins than our **NpGC**. For instance, on Calte101view3, they are lower than ours by 14.94%, 9.08% and 12.89% respectively in terms of ACC. Therefore, our non-parametric framework **NpGC** is more desirable.
8. Methods **MSCIAS** and **BMVC** are with four hyper-parameters, and nevertheless even with the help of so many hyper-parameters, they still do not outperform us in terms of any one metric, which once again highlights our **NpGC**'s superiority.

Running Time

In this subsection we aggregate the running time of all compared approaches to emphasize **NpGC**'s advantages in time expenditure. The comparisons are displayed in Figure 4. According to these results, we can get the conclusions:

1. With the sample size increasing, the running time of all methods gradually increases. These methods can all normally run on Calte101view3. For datasets with medium scale like CCV and STL10, methods **MVSC**, **MCLES**, **PMSC**, **PFSC** and etc become ineffective.

[1]https://www.vision.caltech.edu/datasets/
[2]https://www.ee.columbia.edu/ln/dvmm/CCV/
[3]https://cs.stanford.edu/~acoates/stl10/
[4]https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/
[5]https://www.micc.unifi.it/resources/datasets/e-ytf/
[6]https://www.micc.unifi.it/resources/datasets/e-ytf/

Dataset	Method	Number of Parameters	ACC	NMI	Purity	Fscore	Precision
Calte101view3	MVSC	3	26.02±1.86	15.68±1.29	28.55±1.53	16.47±1.14	15.62±1.00
	AMGL	0	20.61±0.59	12.92±0.78	21.37±0.53	13.59±0.44	10.53±0.28
	MLRSSC	3	17.77±0.00	4.37±0.05	17.77±0.00	16.44±0.02	10.41±0.01
	MSCIAS	4	33.96±1.90	27.42±1.26	38.13±1.52	22.87±1.36	22.22±1.51
	BMVC	4	24.41±0.00	14.42±0.00	27.15±0.00	14.73±0.00	14.56±0.00
	FMR	2	25.45±1.81	16.79±1.45	28.83±1.36	16.02±1.21	16.14±1.22
	mPAC	3	26.95±0.00	16.76±0.00	30.27±0.00	16.28±0.00	17.42±0.00
	MCLES	3	28.05±0.91	17.11±1.10	29.48±1.04	18.07±0.71	15.87±0.77
	PMSC	3	16.42±0.56	4.98±0.36	17.99±0.56	10.85±0.32	9.46±0.12
	PFSC	2	35.41±2.08	26.40±1.46	37.94±1.85	23.72±1.62	24.07±1.65
	FMCNOF	1	21.09±0.00	11.61±0.00	23.24±0.00	16.97±0.00	12.18±0.00
	FPMVS	0	26.47±0.57	19.89±0.75	30.79±0.63	18.40±0.21	14.38±0.15
	SFMC	0	22.66±0.00	15.32±0.00	24.02±0.00	18.63±0.00	11.43±0.00
	MSGL	3	21.29±0.00	10.19±0.00	22.27±0.00	14.96±0.00	10.52±0.00
	PGSC	2	15.93±0.27	8.16±0.38	17.20±0.46	15.79±0.17	9.38±0.03
	UDBGL	3	34.77±0.00	27.90±0.00	37.30±0.00	23.70±0.00	21.87±0.00
	EEOMVC	2	35.16±0.00	24.89±0.00	36.91±0.00	23.25±0.00	22.58±0.00
	Ours	0	**35.55±2.25**	**28.12±1.30**	**38.55±1.80**	**24.29±1.18**	**24.71±1.22**
Dataset	**Method**	**Number of Parameters**	ACC	NMI	Purity	Fscore	Precision
CCV	MVSC	3	N/A				
	AMGL	0	13.60±0.39	12.45±0.58	13.98±0.40	11.12±0.42	7.47±0.29
	MLRSSC	3	11.19±0.86	2.20±0.26	11.46±0.65	10.81±0.21	5.93±0.19
	MSCIAS	4	22.55±0.96	19.11±0.40	26.25±0.75	13.34±0.30	13.03±0.51
	BMVC	4	15.84±0.00	11.46±0.00	17.73±0.00	11.84±0.00	9.54±0.00
	FMR	2	15.95±0.61	12.80±0.23	20.53±0.42	9.75±0.19	10.18±0.18
	mPAC	3	22.72±0.00	16.74±0.00	26.66±0.00	13.25±0.00	13.95±0.00
	MCLES	3	N/A				
	PMSC	3	N/A				
	PFSC	2	N/A				
	FMCNOF	1	14.60±0.00	10.12±0.00	16.36±0.00	11.29±0.00	7.62±0.00
	FPMVS	0	22.39±0.87	16.02±1.01	24.29±1.05	13.21±0.31	11.69±0.57
	SFMC	0	11.03±0.00	2.15±0.00	11.18±0.00	10.84±0.00	5.75±0.00
	MSGL	3	15.70±0.00	11.07±0.00	18.54±0.00	10.71±0.00	7.86±0.00
	PGSC	2	10.57±0.00	0.56±0.00	10.70±0.00	10.84±0.00	5.73±0.00
	UDBGL	3	20.74±0.00	17.34±0.00	22.84±0.00	13.25±0.00	10.14±0.00
	EEOMVC	2	18.12±0.00	13.30±0.00	19.80±0.00	11.65±0.00	8.56±0.00
	Ours	0	**23.39±0.96**	**19.19±0.39**	**26.85±0.80**	**13.53±0.27**	**14.18±0.32**
Dataset	**Method**	**Number of Parameters**	ACC	NMI	Purity	Fscore	Precision
STL10	MVSC	3	N/A				
	AMGL	0	10.75±0.11	0.20±0.02	10.78±0.11	15.69±0.35	10.00±0.00
	MLRSSC	3	N/A				
	MSCIAS	4	72.33±7.23	73.00±3.47	75.31±5.30	65.39±5.25	61.73±6.72
	BMVC	4	62.06±0.00	63.93±0.00	64.22±0.00	55.69±0.00	51.97±0.00
	FMR	2	N/A				
	mPAC	3	N/A				
	MCLES	3	N/A				
	PMSC	3	N/A				
	PFSC	2	N/A				
	FMCNOF	1	23.14±0.00	13.32±0.00	24.30±0.00	15.89±0.00	14.55±0.00
	FPMVS	0	72.49±5.99	69.31±2.58	72.53±5.97	66.31±4.96	57.29±7.44
	SFMC	0	10.06±0.00	0.16±0.00	10.08±0.00	18.17±0.00	9.99±0.00
	MSGL	3	14.18±0.00	1.41±0.00	14.21±0.00	15.81±0.00	10.35±0.00
	PGSC	2	10.05±0.00	0.14±0.00	10.07±0.00	18.17±0.00	9.99±0.00
	UDBGL	3	83.29±0.00	**88.99±0.00**	87.66±0.00	83.87±0.00	78.30±0.00
	EEOMVC	2	79.11±0.00	59.56±0.00	79.11±0.00	63.15±0.00	63.06±0.00
	Ours	0	**87.46±8.26**	88.53±2.90	**90.20±5.79**	**86.23±6.50**	**82.96±9.33**

Table 1: Clustering results on real datasets Calte101view3, CCV and STL10 (mean±std). N/A represents that algorithm can not normally work on this dataset.

Dataset	**Method**	**Number of Parameters**	**Metric**				
			ACC	NMI	Purity	Fscore	Precision
VGGFace100	MVSC	3			N/A		
	AMGL	0			N/A		
	MLRSSC	3			N/A		
	MSCIAS	4			N/A		
	BMVC	4	6.17±0.00	14.26±0.00	7.15±0.00	2.77±0.00	2.66±0.00
	FMR	2			N/A		
	mPAC	3			N/A		
	MCLES	3			N/A		
	PMSC	3			N/A		
	PFSC	2			N/A		
	FMCNOF	1	3.17±0.00	5.80±0.00	3.25±0.00	2.38±0.00	1.27±0.00
	FPMVS	0	4.93±0.13	9.71±0.25	5.07±0.14	2.82±0.01	1.58±0.01
	SFMC	0			N/A		
	MSGL	3	4.70±0.00	9.32±0.00	5.22±0.00	2.10±0.00	1.35±0.00
	PGSC	2			N/A		
	UDBGL	3			N/A		
	EEOMVC	2	7.68±0.00	14.04±0.00	8.54±0.00	3.11±0.00	2.71±0.00
	Ours	0	**9.28±0.32**	**16.95±0.33**	**10.25±0.31**	**3.84±0.13**	**3.89±0.13**
Dataset	**Method**	**Number of Parameters**	**Metric**				
			ACC	NMI	Purity	Fscore	Precision
YTF20	MVSC	3			N/A		
	AMGL	0			N/A		
	MLRSSC	3			N/A		
	MSCIAS	4			N/A		
	BMVC	4	57.39±0.00	70.65±0.00	62.76±0.00	49.04±0.00	47.13±0.00
	FMR	2			N/A		
	mPAC	3			N/A		
	MCLES	3			N/A		
	PMSC	3			N/A		
	PFSC	2			N/A		
	FMCNOF	1	39.56±0.00	48.49±0.00	45.68±0.00	29.35±0.00	24.26±0.00
	FPMVS	0	61.12±3.46	72.58±2.23	63.92±3.83	54.37±4.76	46.84±6.53
	SFMC	0			N/A		
	MSGL	3	60.05±0.00	69.15±0.00	64.25±0.00	42.89±0.00	33.38±0.00
	PGSC	2			N/A		
	UDBGL	3	**68.98±0.00**	**78.66±0.00**	**74.02±0.00**	**61.40±0.00**	56.19±0.00
	EEOMVC	2			N/A		
	Ours	0	65.76±5.13	77.09±1.90	72.53±3.43	60.64±4.68	**58.22±5.64**
Dataset	**Method**	**Number of Parameters**	**Metric**				
			ACC	NMI	Purity	Fscore	Precision
YTF50	MVSC	3			N/A		
	AMGL	0			N/A		
	MLRSSC	3			N/A		
	MSCIAS	4			N/A		
	BMVC	4	65.99±0.00	81.87±0.00	**73.64±0.00**	57.08±0.00	53.44±0.00
	FMR	2			N/A		
	mPAC	3			N/A		
	MCLES	3			N/A		
	PMSC	3			N/A		
	PFSC	2			N/A		
	FMCNOF	1	26.53±0.00	44.98±0.00	27.82±0.00	16.04±0.00	10.44±0.00
	FPMVS	0	62.56±2.57	80.32±1.05	64.34±2.43	54.86±3.38	44.93±4.23
	SFMC	0			N/A		
	MSGL	3	44.33±0.00	61.61±0.00	51.24±0.00	24.74±0.00	17.03±0.00
	PGSC	2			N/A		
	UDBGL	3	64.11±0.00	80.73±0.00	70.71±0.00	54.36±0.00	48.44±0.00
	EEOMVC	2			N/A		
	Ours	0	**66.13±3.01**	**83.08±0.87**	73.36±2.00	**59.60±2.94**	**55.49±3.74**

Table 2: Clustering results on real datasets VGGFace100, YTF20 and YTF50 (mean±std). N/A represents that algorithm can not normally work on this dataset.

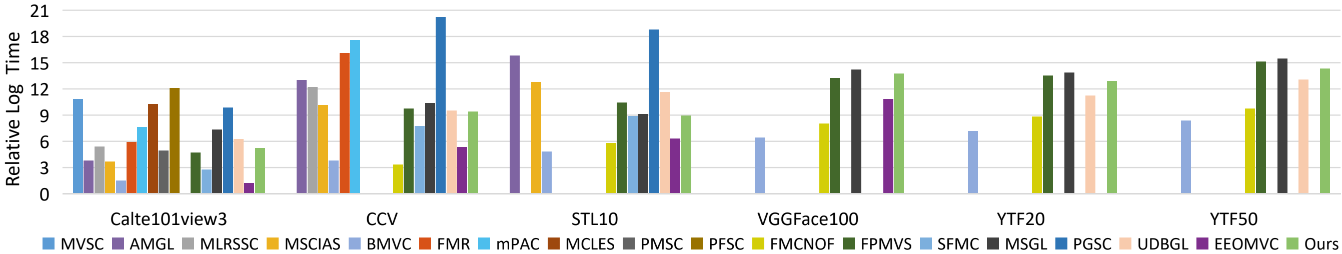


Figure 4: Running time of 18 algorithms.

Dataset	Metric	ACC	NMI	Purity	Fscore	Precision
Calte101view3	VR	33.96±1.50	27.44±1.13	37.42±1.23	23.45±0.81	23.89±0.89
	VUR	32.95±3.37	26.00±1.07	35.82±2.49	22.26±3.51	22.56±4.43
	Ours	**35.55±2.25**	**28.12±1.30**	**38.55±1.80**	**24.29±1.18**	**24.71±1.22**
CCV	VR	22.81±0.92	19.05±0.40	25.90±0.74	13.34±0.30	13.99±0.32
	VUR	23.01±0.75	19.01±0.38	25.95±0.66	13.18±0.28	13.88±0.32
	Ours	**23.39±0.96**	**19.19±0.39**	**26.85±0.80**	**13.53±0.27**	**14.18±0.32**
STL10	VR	83.99±7.66	86.97±2.79	87.97±5.43	84.38±6.05	79.61±8.46
	VUR	86.13±3.91	73.62±2.07	87.22±3.08	77.16±3.09	76.79±3.98
	Ours	**87.46±8.26**	**88.53±2.90**	**90.20±5.79**	**86.23±6.50**	**82.96±9.33**
VGGFace100	VR	9.01±0.41	16.52±0.35	9.96±0.38	3.23±0.16	3.28±0.16
	VUR	8.44±0.31	16.18±0.28	9.29±0.34	3.03±0.10	3.16±0.10
	Ours	**9.28±0.32**	**16.95±0.33**	**10.25±0.31**	**3.84±0.13**	**3.89±0.13**
YTF20	VR	64.97±5.22	76.05±1.89	71.66±3.62	59.57±4.79	57.32±5.70
	VUR	64.32±5.33	75.98±1.70	71.23±3.26	59.09±4.33	57.17±5.14
	Ours	**65.76±5.13**	**77.09±1.90**	**72.53±3.43**	**60.64±4.68**	**58.22±5.64**
YTF50	VR	65.32±3.37	81.39±1.07	72.00±2.49	58.31±3.51	54.36±4.43
	VUR	64.99±3.19	81.10±0.96	71.80±2.14	57.94±3.46	53.79±4.29
	Ours	**66.13±3.01**	**83.08±0.87**	**73.36±2.00**	**59.60±2.94**	**55.49±3.74**

Table 3: Ablation results for only using $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ or $\mathbf{B}$.

2. **PFSC** is the most sensitive to the sample size, which is mainly caused by its quartic time and square space complexities. Cubic computational overhead also leads to **MVSC**, **AMGL**, **MLRSSC**, **FMR**, **PMSC** and etc encountering a relatively slow running speed.
3. Methods **MVSC**, **MLRSSC**, **FMR**, **mPAC**, **PMSC**, **MCLES** and etc are generally slower than **FPMVS**, **SFMC**, **FMCNOF**, **EEOMVC**, **UDBGL** and ours. This is mainly because the formers adopt a subspace strategy to establish the similarity relationship, and generally need to construct the full graph.
4. **NpGC** typically works well on these multi-view datasets within an acceptable time range, including datasets with small, middle, and large scales respectively. This indicates that **NpGC** is with wider applicability.
5. Compared to large-scale oriented methods **BMVC**, **FPMVS**, **MSGL**, **FMCNOF**, **UDBGL** and etc, **NpGC** can provide competitive running speed. Specially, **NpGC** needs less time than **MSGL** consistently.
6. Although **BMVC** and **FMCNOF** are slightly faster than ours, the plain code and random sampling operations they adopted are not conducive to sufficiently exploiting consistency and complementarity among heterogeneous data, producing sub-optimal results.

Ablation Study

In the paper we employ two types of anchors, view-related $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ and view-unrelated $\mathbf{B}$, to concurrently explore exclusive and common characteristics among views so as to form high-quality representations. To reveal the effectiveness of each part, we organize some ablation studies, as shown in Table 3 where VR and VUR are the results using only $\{\mathbf{B}^{(r)}\}_{r=1}^{v}$ and $\mathbf{B}$ respectively. As seen, we can combine $\mathbf{B}^{(r)}$ and $\mathbf{B}$'s information to achieve better results.

Concluding Remarks

We proposed a non-parametric framework NpGC to deal with multi-view clustering problems more practically. It adopts two kinds of anchors to simultaneously mine exclusive and common characteristics among views. Unlike previous algorithms integrating multi-view information at graph level, it learns directly at anchor level, which allows the original diversity to be well maintained. Also, the consensus bipartite graph directly links these two kinds of anchors so that they can negotiate mutually to improve their discrimination. Experiments have shown that it produces competitive results, even on the dataset with over 120000 samples. In the future, we will strive to design appropriate weighting schemes for anchors to further increase the performance.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2022ZD0209103); in part by the National Natural Science Foundation of China (No. 62325604 and 62276271); in part by the Hunan Provincial Natural Science Foundation of China (No. 2021JJ30779).

References

Bezdek, J. C.; and Hathaway, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations*, 11(4): 351–368.

Brbić, M.; and Kopriva, I. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73: 247–258.

Chen, M.-S.; Huang, L.; Wang, C.-D.; and Huang, D. 2020. Multi-view clustering in latent embedding space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 3513–3520.

Du, S.; Liu, Z.; Chen, Z.; Yang, W.; and Wang, S. 2021. Differentiable bi-sparse multi-view co-clustering. *IEEE Transactions on Signal Processing*, 69: 4623–4636.

Fang, S.-G.; Huang, D.; Cai, X.-S.; Wang, C.-D.; He, C.; and Tang, Y. 2023. Efficient multi-view clustering via unified and discrete bipartite graph learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Fu, S.; Cao, Q.; Lei, Y.; Zhong, Y.; Zhan, Y.; and You, X. 2023. Few-Shot Learning With Dynamic Graph Structure Preserving. *IEEE Transactions on Industrial Informatics*.

Fu, S.; Liu, B.; Liu, W.; Zou, B.; You, X.; Peng, Q.; and Jing, X.-Y. 2022. Adaptive multi-scale transductive information propagation for few-shot learning. *Knowledge-Based Systems*, 249: 108979.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, 4238–4246.

Huang, S.; Kang, Z.; and Xu, Z. 2020. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, 97: 107015.

Huang, S.; Xu, Z.; Tsang, I. W.; and Kang, Z. 2020. Auto-weighted multi-view co-clustering with bipartite graphs. *Information Sciences*, 512: 18–30.

Kang, Z.; Guo, Z.; Huang, S.; Wang, S.; Chen, W.; Su, Y.; and Xu, Z. 2019. Multiple partitions aligned clustering. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2701–2707.

Kang, Z.; Lin, Z.; Zhu, X.; and Xu, W. 2022. Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview. *IEEE transactions on cybernetics*, 52(9): 8976–8986.

Kang, Z.; Lu, X.; Yi, J.; and Xu, Z. 2018. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2312–2318.

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

Khan, A.; and Maji, P. 2021. Approximate Graph Laplacians for Multimodal Data Clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(03): 798–813.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized multi-view spectral clustering. *Advances in neural information processing systems*, 24.

Li, R.; Zhang, C.; Hu, Q.; and et al. 2019. Flexible Multi-View Representation Learning for Subspace Clustering. In *IJCAI*, 2916–2922.

Li, X.; Sun, Y.; Sun, Q.; and Ren, Z. 2022a. Consensus Cluster Center Guided Latent Multi-kernel Clustering. *IEEE Transactions on Circuits and Systems for Video Technology*.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2022b. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE transactions on pattern analysis and machine intelligence*, 44(1): 330–344.

Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023a. Knowledge Graph Contrastive Learning Based on Relation-Symmetrical Structure. *IEEE Transactions on Knowledge and Data Engineering*.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2023b. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1559–1568.

Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7576–7584.

Liu, X.; Zhou, S.; Liu, L.; and et al. 2021. Localized simple multiple kernel k-means. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9293–9301.

Luong, K.; Nayak, R.; Balasubramaniam, T.; and Bashar, M. A. 2022. Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering. *Pattern Recognition*, 131: 108815.

Lv, J.; Kang, Z.; Wang, B.; Ji, L.; and Xu, Z. 2021. Multi-view subspace clustering via partition fusion. *Information Sciences*, 560: 410–423.

Nie, F.; Li, J.; and Li, X. 2016. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 1881–1887.

Nie, F.; Shi, S.; and Li, X. 2020. Auto-weighted multi-view co-clustering via fast matrix factorization. *Pattern Recognition*, 102: 107207.

Qin, Y.; Wu, H.; Zhang, X.; and Feng, G. 2021. Semi-supervised structured subspace learning for multi-view clustering. *IEEE Transactions on Image Processing*, 31: 1–14.

Shi, S.; Nie, F.; Wang, R.; and Li, X. 2021. Fast multi-view clustering via prototype graph. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 443–455.

Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; Zhu, E.; and Cheng, J. 2021. Deep fusion clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9978–9987.

Wan, X.; Liu, J.; Liang, W.; Liu, X.; Wen, Y.; and Zhu, E. 2022. Continual multi-view clustering. In *Proceedings of the 30th ACM International Conference on Multimedia*, 3676–3684.

Wan, X.; Liu, X.; Liu, J.; Wang, S.; Wen, Y.; Liang, W.; Zhu, E.; Liu, Z.; and Zhou, L. 2023. Auto-Weighted Multi-View Clustering for Large-Scale Data. 10078–10086.

Wang, H.; Yang, Y.; and Liu, B. 2020. GMC: Graph-Based Multi-View Clustering. *IEEE Transactions on Knowledge & Data Engineering*, 32(06): 1116–1129.

Wang, J.; Tang, C.; Wan, Z.; Zhang, W.; Sun, K.; and Zomaya, A. Y. 2023. Efficient and Effective One-Step Multiview Clustering. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, Q.; Tao, Z.; Xia, W.; Gao, Q.; Cao, X.; and Jiao, L. 2022a. Adversarial multiview clustering networks with adaptive fusion. *IEEE transactions on neural networks and learning systems*.

Wang, S.; Liu, X.; Liu, L.; Zhou, S.; and Zhu, E. 2021. Late fusion multiple kernel clustering with proxy graph refinement. *IEEE Transactions on Neural Networks and Learning Systems*.

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 *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3778–3784.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2022b. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Wen, J.; Liu, C.; Deng, S.; Liu, Y.; Fei, L.; Yan, K.; and Xu, Y. 2023. Deep Double Incomplete Multi-View Multi-Label Learning With Incomplete Labels and Missing Views. *IEEE Transactions on Neural Networks and Learning Systems*.

Wen, J.; Zhang, Z.; Fei, L.; Zhang, B.; Xu, Y.; Zhang, Z.; and Li, J. 2022. A survey on incomplete multiview clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(2): 1136–1149.

Wu, H.; Huang, S.; Tang, C.; Zhang, Y.; and Lv, J. 2023. Pure graph-guided multi-view subspace clustering. *Pattern Recognition*, 136: 109187.

Xia, W.; Wang, Q.; Gao, Q.; Zhang, X.; and Gao, X. 2021a. Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, 24: 3182–3192.

Xia, W.; Wang, T.; Gao, Q.; Yang, M.; and Gao, X. 2023. Graph embedding contrastive multi-modal representation learning for clustering. *IEEE Transactions on Image Processing*, 32: 1170–1183.

Xia, W.; Zhang, X.; Gao, Q.; Shu, X.; Han, J.; and Gao, X. 2021b. Multiview subspace clustering by an enhanced tensor nuclear norm. *IEEE Transactions on cybernetics*, 52(9): 8962–8975.

Yang, B.; Zhang, X.; Nie, F.; Wang, F.; Yu, W.; and Wang, R. 2020. Fast multi-view clustering via nonnegative and orthogonal factorization. *IEEE Transactions on Image Processing*, 30: 2575–2586.

Yang, B.; Zhang, X.; Nie, F.; Wang, F.; Yu, W.; and Wang, R. 2021a. Fast Multi-View Clustering via Nonnegative and Orthogonal Factorization. *IEEE Transactions on Image Processing*, 30: 2575–2586.

Yang, H.; Gao, Q.; Xia, W.; Yang, M.; and Gao, X. 2022a. Multiview spectral clustering with bipartite graph. *IEEE Transactions on Image Processing*, 31: 3591–3605.

Yang, Y.; Guan, Z.; Wang, Z.; Zhao, W.; Xu, C.; Lu, W.; and Huang, J. 2022b. Self-supervised Heterogeneous Graph Pre-training Based on Structural Clustering. In *NeurIPS*, volume 35, 16962–16974.

Yang, Y.; Yang, J.-Q.; Bao, R.; Zhan, D.-C.; Zhu, H.; Gao, X.-R.; Xiong, H.; and Yang, J. 2021b. Corporate relative valuation using heterogeneous multi-modal graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 211–224.

Yu, S.; Liu, S.; Wang, S.; Tang, C.; Luo, Z.; Liu, X.; and Zhu, E. 2023a. Sparse Low-Rank Multi-View Subspace Clustering With Consensus Anchors and Unified Bipartite Graph. *IEEE Transactions on Neural Networks and Learning Systems*.

Yu, S.; Wang, S.; Wen, Y.; Wang, Z.; Luo, Z.; Zhu, E.; and Liu, X. 2023b. How to Construct Corresponding Anchors for Incomplete Multiview Clustering. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, C.; Wang, S.; Liu, J.; Zhou, S.; Zhang, P.; Liu, X.; Zhu, E.; and Zhang, C. 2021. Multi-view clustering via deep matrix factorization and partition alignment. In *Proceedings of the 29th ACM international conference on multimedia*, 4156–4164.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2019. Binary Multi-View Clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1774–1782.