



Research article

Semi-supervised graph regularized concept factorization with the class-driven constraint for image representation

Yuelin Gao¹, Huirong Li^{2,3,*}, Yani Zhou⁴ and Yijun Chen⁵

¹ School of Mathematics and Information Science, North Minzu University, Yinchuan, 750021, China

² School of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China

³ Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi Province, Shangluo 726000, China

⁴ School of Health Management, Shangluo University, Shangluo 726000, China

⁵ Library, Xian Aeronautical University, Xi'an 710077, China

* **Correspondence:** Email: hrli@slxy.edu.cn.

Abstract: As a popular dimensionality reduction technique, concept factorization (CF) has been widely applied in image clustering. However, CF fails to extract the intrinsic structure of data space and does not utilize the label information. In this paper, a new semi-supervised graph regularized CF (SGCF) method is proposed, which makes full use of the limited label information and the graph regularization to improve the algorithm of clustering performance. Particularly, SGCF associates the class label information of data points with their new representations by using the class-driven constraint, and this constraint forces the new representations of data points to be more similar within the same class while different between classes. Furthermore, SGCF extracts the geometric structure of the data space by incorporating graph regularization. SGCF not only reveals the geometrical structure of the data space, but also takes into the limited label information account. We drive an efficient multiplicative update algorithm for SGCF to solve the optimization, and analyze the proposed SGCF method in terms of the convergence and computational complexity. Clustering experiments show the effectiveness of the SGCF method in comparison to other state-of-the-art methods.

Keywords: concept factorization; semi-supervised learning; class-driven constraint; label information

Mathematics Subject Classification: 15A23

1. Introduction

In real-world applications, the image clustering problem has become an important research hotspot in the areas of machine learning and pattern recognition due to it enabled knowledge discovery. With the rapid development of modern information technology, we are usually faced with high-dimensional data generated in the real world [1, 2]. In addition, this high-dimensional data not only increases the calculation and storage but also often encounters some problems such as “curse of dimension” and “small sample size”, which will bring great difficulties to the subsequent processing of data [3]. In general, data representation plays a crucial role in the field of computer vision in order to discover the important underlying structures and useful information in high-dimensional data. Matrix factorization (MF) is one of these data representation techniques, and typical methods include principal component analysis (PCA) [4], singular value decomposition (SVD) [5], nonnegative matrix factorization (NMF) [6, 7] and low rank representation (LRR) [8, 9]. NMF is a popular one of the MF methods, which obtains a parts-based representation of original data space. The great advantage of this parts-based representation have been successfully applied in various areas, such as clustering and classification [10–12], face and object recognition [13, 14], gene expression analysis [15] and others. However, NMF still has some intrinsic drawbacks [16, 17]. For example, it is only applicable to nonnegative data matrix and cannot utilize the power of kernelization, and it fails to extract the geometric structure of data space and cannot make use of the label information.

In order to overcome these above shortcomings, many improved NMF algorithms have been successfully proposed in recent years. For example, Xu and Gong proposed a concept factorization (CF) method [18]. In CF, each cluster is represented by a linear combination of the data points, and each data point is linearly represented by the cluster centers. Although the CF method can be applied in any transformed space, it is difficult to discover the local geometric structures of the data space. Therefore, several graph regularized CF methods have been proposed to handle this issue, such as locally consistent CF (LCCF) [19], local coordinate CF (LCF) [20], adaptive graph guided CF (AGCF) [21], and so on. Specifically, LCCF extracts the intrinsic geometrical structures of data space by using a k-nearest neighbor graph, while LCF preserves the sparsity and locality simultaneously by introducing a local coordinate constraint and AGCF captures the local relationships of the data points by using the adaptive graph regularization constraint. However, most algorithms construct graphs directly on the original data, which can lead to sensitivity to noise [22]. Many studies have employed adaptive graphs to solve these problems [23–25], but they mainly focus primarily on the problem of data dimensionality reduction.

Undoubtedly, the improved CF models mentioned above are unsupervised learning methods, and they fail to utilize the limited label information. Using a small amount of label information of the data to improve the performance of the algorithms has become one of the hotspots of machine learning in current studies. Many machine learning experts have shown that a significant improvement can be achieved in learning accuracy when a small amount of the labeled data is used in conjunction with the large amounts of the unlabeled data [26–32]. In order to make better use of the label information, several works incorporated the limited label information into CF. For examples, Liu et al. incorporated the label constraints into NMF and CF, respectively, and proposed both constrained nonnegative matrix factorization (CNMF) [14] and constrained concept factorization (CCF) [33]. The purpose of these constraints is that the samples from the same class should be merged together in the new representation

space. Babaee et al proposed a discriminative NMF (DNMF) [27] by using the label information of a small amount of data as a discriminative constraint, which enforces the samples with the same label information to be aligned on the same axis in the new representation instead of a single point. Li et al. proposed a class-driven CF by utilizing the class-driven constraint, which forces the representations of samples to be more similar within the same class while different between other classes. In addition, some further research of the CF has also been proposed [32, 34–36] based on different constraints in the past few years.

In this work, we incorporate the class-driven constraint and the graph regularization into CF, and propose a new semi-supervised graph regularized CF (SGCF) algorithm for the clustering application. Specifically, the main contributions of this work can be outlined as follows:

- In our proposed SGCF, we incorporate the limited label information and the graph regularization into CF as the additional constraints to enhance the performance of the clustering task.
- SGCF is a semi-supervised learning method, which associates the class labels of samples with their representations by using a class-driven constraint, and also extracts the intrinsic geometrical structure of the data space by introducing the graph regularization.
- The multiplicative update algorithm is used to optimize the objective function of SGCF. Meanwhile, the multiplicative update rules and the computational complexity of SGCF are derived, and its convergence is also proved.
- Extensive experiments on four real-word image data sets show that the effectiveness of the proposed SGCF when compared with the state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 briefly reviews some methods that are closely related with our research, including NMF, CF, GNMF and the class-driven constraint. We then introduce our proposed SGCF model and its optimization method in Section 3, and its convergence proof is provided in Section 4. Next, we conduct extensive experiments in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

In this section, we briefly introduce the basic concepts of NMF, CF, GNMF and the class-driven constraint, which are closely related with our proposed SGCF method.

2.1. NMF and CF

Given a data matrix $\mathbf{X} = [x_1, x_2, \dots, x_n] \in \mathbf{R}_+^{m \times n}$, where $x_i \in \mathbf{R}_+^m (1 \leq i \leq n)$ represents a nonnegative data point. NMF aims to decompose \mathbf{X} into two matrix factors, i.e., the basis matrix $\mathbf{U} \in \mathbf{R}_+^{m \times k}$ and the coefficient matrix $\mathbf{V} \in \mathbf{R}_+^{n \times k}$, which minimize the following optimization problem [6, 7]:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} O &= \|\mathbf{X} - \mathbf{UV}^T\|_F^2, \\ \text{s.t. } \mathbf{U} &\geq 0, \mathbf{V} \geq 0, \end{aligned} \quad (2.1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of the matrix. This formulation can be viewed column by column as follows:

$$x_j = \sum_{i=1}^k v_{ji} u_i, \quad (2.2)$$

where x_j can be approximated by a linear combination of \mathbf{U} with the weight matrix \mathbf{V} .

However, NMF cannot utilize the power of kernelization. Therefore, Xu and Gong proposed CF, which can be used in any data representation space [18]. The CF is an extension of NMF in that each basis vector u_j is represented by a linear combination of the samples x_i :

$$u_j = \sum_{i=1}^n w_{ij} x_i,$$

where $w_{ij} \geq 0$. Therefore, let $\mathbf{W} = [w_{ij}] \in R^{n \times k}$, and CF can be expressed as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{V}} O &= \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|_F^2 \\ \text{s.t. } &\mathbf{W} \geq 0, \mathbf{V} \geq 0. \end{aligned} \quad (2.3)$$

In this way, CF can easily be performed in any data representation space, such as Hilbert space. Some detailed review can be found in [18–20, 35].

2.2. Graph-regularized nonnegative matrix factorization (GNMF)

NMF cannot detect the intrinsic structure of the original data space, which is essential to the image representation. For example, Cai et al. proposed the graph-regularized NMF (GNMF) method, which removes this restriction by adding a regularizer based on the manifold assumption that if two samples x_i, x_j are similar, then their weighted coefficient v_i, v_j are similar [37]. The weight s_{ij} is used to measure the similarity of two samples x_i, x_j . There are many choices to construct the weight matrix \mathbf{S} on the graph, such as dot-product weighting, heat kernel weighting, 0-1 weighting and so on [38]. Then, $Tr(\mathbf{V}^T \mathbf{L} \mathbf{V})$ can be used to preserve the local manifold structure of the data space, so the objective function of GNMF is given as follows:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} O_F &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \alpha Tr(\mathbf{V}^T \mathbf{L} \mathbf{V}), \\ \text{s.t. } &\mathbf{U} \geq 0, \mathbf{V} \geq 0, \end{aligned} \quad (2.4)$$

where $\alpha > 0$, $Tr(\cdot)$ denotes the trace of a matrix. $\mathbf{L} = \mathbf{H} - \mathbf{S}$ is the graph Laplacian matrix [39], \mathbf{S} is the weight matrix and \mathbf{H} is diagonal matrix with the i th diagonal element $\mathbf{H}_{ii} = \sum_j \mathbf{S}_{ij}$. In general, the matrix \mathbf{S} is introduced as follows:

$$\mathbf{S}_{ij} = \begin{cases} \frac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i \in N_p(x_j) \text{ or } x_j \in N_p(x_i), \\ 0 & \text{otherwise,} \end{cases} \quad (2.5)$$

where $N_p(x_i)$ represents p -nearest neighbors set of x_i .

2.3. Class-driven constraint

Consider a dataset consisting of n samples $\mathbf{X} = [x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n]$, among which the first l samples x_1, x_2, \dots, x_l are labeled with one class form c categories and the rest of $n - l$ samples x_{l+1}, \dots, x_n are unlabeled samples. In order to learn a discriminative basis set $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_k] \in \mathbf{R}_+^{m \times k}$, where \mathbf{U}_i is able to sparsely represent the i -th class well but not others, r denotes the number of basis vectors of each subset and $k = r \times c$.

Suppose $\mathbf{V} = [v_1, v_2, \dots, v_n]^T \in \mathbf{R}^{n \times k}$ denotes the representation matrix in Eq (2.3) and $\mathbf{D} = [d_{b_1}, d_{b_2}, \dots, d_{b_n}]^T \in \mathbf{R}^{n \times k}$ denotes the indicator matrix of the inhomogeneous representation. Our target is to make use of class label information to learn discriminative basis vectors, which represent better for their own classes while worse for the others. Therefore, we expect that approximated part-based representation v_j for sample x_j with label b_j will satisfy the following conditions [40]:

$$d_{b_j}^T v_j = 0, \quad (2.6)$$

where d_{b_j} chooses the inhomogeneous representation coefficients of v_j , i.e., coefficients corresponding to basis vectors other than \mathbf{U}_{b_j} . There are n samples, suppose the discriminative basis set $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]$, $\mathbf{U}_i \in \mathbf{R}^{m \times 2}$ (i.e., $k = 6$), where x_1 belongs to the first class, x_2 belongs to the second class, x_3 belongs to the third class and the other $n - 3$ are unlabeled samples. Thus, the indicator matrix $\mathbf{D}^T = [d_{b_1}, d_{b_2}, d_{b_3}, \dots, d_{b_n}]$ can be defined as

$$\begin{pmatrix} 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \end{pmatrix},$$

$\underbrace{\hspace{10em}}_l$
 $\underbrace{\hspace{10em}}_{n-l}$

where if x_j (such as x_4) is the unlabeled sample, we set all the elements in d_{b_j} (such as d_{b_4}) to zero. We refer to Eq (2.6) as the class-driven constraint.

Li et al. incorporated the class-driven constraint Eq (2.6) into CF and proposed a class-driven CF (CDCF) as follows [41]:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{V}} O &= \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|_F^2 + \beta \text{Tr}(\mathbf{D}\mathbf{V}^T) \\ \text{s.t. } & \mathbf{W} \geq 0, \mathbf{V} \geq 0. \end{aligned} \quad (2.7)$$

The objective function is solved by the iterative updating method, and the updating rules of \mathbf{W} and \mathbf{V} are:

$$w_{ik} \leftarrow w_{ik} \frac{(\mathbf{K}\mathbf{V})_{ik}}{(\mathbf{K}\mathbf{W}\mathbf{V}^T\mathbf{V})_{ik}}, \quad (2.8)$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{K}\mathbf{W})_{jk}}{(\mathbf{V}\mathbf{W}^T\mathbf{K}\mathbf{W} + (\beta\mathbf{D})/2)_{jk}}. \quad (2.9)$$

3. SGCF

3.1. The objective function

CDCF is a semi-supervised learning method, but it fails to extract the geometrical structure of data space [41]. GNMF adds the manifold regularization to NMF [37]. LCF considers the locality constraints in revealing the underlying concepts, but it ignores the manifold structure in the data space [20]. Therefore, in order to reveal the latent concepts consistent with manifold geometrical structure, we will incorporate this manifold regularization $Tr(V^T L V)$ to CDCF and obtain the proposed SGCF approach formulated as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{V}} O &= \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|_F^2 + \alpha Tr(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \beta Tr(\mathbf{D}\mathbf{V}^T) \\ s.t. \quad &\mathbf{W} \geq 0, \mathbf{V} \geq 0, \end{aligned} \quad (3.1)$$

where α, β are nonnegative balance factors. The key novelty of SGCF is that SGCF not only captures the intrinsic geometrical structure of data space by using graph regularization but also considers the available label information by incorporating the class-driven constraint. Then, the discriminative ability of SGCF is greatly enhanced in clustering tasks.

3.2. Optimization

Undoubtedly, the objective function of the proposed SGCF method in Eq (3.1) is nonconvex in all matrix variables \mathbf{W} and \mathbf{V} together, while it is convex with respect to the single matrix variables \mathbf{W} or \mathbf{V} . In this situation, it is very hard to find the global optimal solution in Eq (3.1). However, a local minimum can be obtained for the objective function of SGCF by using the multiplicative update rules [28, 37].

We need to minimize the objective function O defined in Eq (3.1) to get its local minima, and the objective function of SGCF can be rewritten as:

$$\begin{aligned} O &= Tr((\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T)^T (\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T)) + \alpha Tr(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \beta Tr(\mathbf{D}\mathbf{V}^T) \\ &= Tr(\mathbf{K}) - 2Tr(\mathbf{W}^T \mathbf{K} \mathbf{V}) + Tr(\mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T \mathbf{V}) + \alpha Tr(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \beta Tr(\mathbf{D}\mathbf{V}^T), \end{aligned} \quad (3.2)$$

where $\mathbf{K} = \mathbf{X}^T \mathbf{X}$. Specifically, assuming that $\Phi = [\phi_{ik}]$ and $\Psi = [\psi_{jk}]$ are two Lagrangian multipliers, then the corresponding Lagrangian function L can be presented below:

$$\begin{aligned} L &= Tr(\mathbf{K}) - 2Tr(\mathbf{W}^T \mathbf{K} \mathbf{V}) + Tr(\mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T \mathbf{V}) \\ &\quad + \alpha Tr(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \beta Tr(\mathbf{D}\mathbf{V}^T) + Tr(\Phi \mathbf{W}^T) + Tr(\Psi \mathbf{V}^T). \end{aligned} \quad (3.3)$$

The partial derivatives of L with respect to \mathbf{W} and \mathbf{V} , respectively, are:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{W}} &= -2\mathbf{K}\mathbf{V} + 2\mathbf{K}\mathbf{W}\mathbf{V}^T \mathbf{V} + \Phi, \\ \frac{\partial L}{\partial \mathbf{V}} &= -2\mathbf{K}\mathbf{W} + 2\mathbf{V}\mathbf{W}^T \mathbf{K} \mathbf{W} + 2\alpha \mathbf{L}\mathbf{V} + \beta \mathbf{D} + \Psi. \end{aligned}$$

Letting the derivatives be zero and using the Karush-Kuhn-Tucker (KKT) condition $\phi_{ik} w_{ik} = 0$ and $\psi_{jk} v_{jk} = 0$, we have:

$$-(\mathbf{K}\mathbf{V})_{ik} w_{ik} + (\mathbf{K}\mathbf{W}\mathbf{V}^T \mathbf{V})_{ik} w_{ik} = 0,$$

$$-(\mathbf{KW})_{jk}v_{jk} + (\mathbf{VW}^T\mathbf{KW})_{jk}v_{jk} + \alpha((\mathbf{H} - \mathbf{S})\mathbf{V})_{jk}v_{jk} + \frac{1}{2}(\beta\mathbf{D})_{jk}v_{jk} = 0.$$

Therefore, we have the following updating rules for \mathbf{W} and \mathbf{V} :

$$w_{ik} \leftarrow w_{ik} \frac{(\mathbf{KV})_{ik}}{(\mathbf{KWV}^T\mathbf{V})_{ik}}, \quad (3.4)$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{KW} + \alpha\mathbf{SV})_{jk}}{(\mathbf{VW}^T\mathbf{KW} + \alpha\mathbf{HV} + (\beta\mathbf{D})/2)_{jk}}. \quad (3.5)$$

From Eq (3.1), we can find that if the indicator matrix \mathbf{D} is a $n \times k$ null matrix or $\beta = 0$, SGCF reduces to LCCF [19], if $\alpha = 0$, SGCF reduces to CDCF [41] and if $\alpha = \beta = 0$, SGCF reduces to CF [18]. The procedure of SGCF is summarized in Algorithm 1.

Algorithm 1. The description of the proposed SGCF algorithm.

Input: Data matrix \mathbf{X} for c classes, graph Laplacian matrix \mathbf{L} , the label indicator matrix D , the parameters α and β .

Initialization: Randomly initialize nonnegative matrices \mathbf{W} , \mathbf{V} .

Repeat

1) Update \mathbf{W} by rule (3.4);

2) Update \mathbf{V} by rule (3.5);

Until the objective value of O does not change or obtain the maximum number of iterations.

Output: Basis matrices \mathbf{W} , \mathbf{V} .

The solution of the optimization problem Eq (3.1) is not unique. It is easy to verify that if \mathbf{W} and \mathbf{V} are a solution of the optimization problem Eq (3.1), then \mathbf{WS} and \mathbf{VS}^{-1} will also be a solution of the optimization problem Eq (3.1) for any positive diagonal matrix \mathbf{S} . Therefore, to make the solution unique, we can further require that $w^T\mathbf{K}w = 1$, where w is the column vector of \mathbf{W} and \mathbf{WV}^T does not change [19,33,41]. So, \mathbf{W} and \mathbf{V} should be updated as following:

$$\begin{aligned} \mathbf{V} &\leftarrow \mathbf{V}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2}, \\ \mathbf{W} &\leftarrow \mathbf{W}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{-1/2}. \end{aligned}$$

3.3. Computational complexity analysis

In this section, we discuss the computational cost of SGCF compared to standard CF and LCCF. Without loss of generality, the big O notation is used to represent the complexity of algorithms. The number of three arithmetic operations including addition, multiplication and division are counted for each update step of NMF, CF, LCCF and proposed SGCF and the results are shown in Table 1 (N : the number of data points; M : the number of features; K : the number of factors; p : the number of nearest neighbors, $p \ll N$).

Table 1. Computational operation counts for each iteration in NMF, CF, LCCF and SGCF.

| Methods | Addition | Multiplication | Division | Overall |
|-------------|-----------------------------|---------------------------------|------------|-----------|
| <i>NMF</i> | $2MNK + 2(M + N)K^2$ | $2MNK + 2(M + N)K^2 + (M + N)K$ | $(M + N)K$ | $O(MNK)$ |
| <i>CF</i> | $4N^2K + 4NK^2$ | $4N^2K + 4NK^2 + 2NK$ | $2NK$ | $O(N^2K)$ |
| <i>LCCF</i> | $4N^2K + 4NK^2 + N(p + 3)K$ | $4N^2K + 4NK^2 + N(p + 3)K$ | $2NK$ | $O(N^2K)$ |
| <i>SGCF</i> | $4N^2K + 4NK^2 + N(p + 4)K$ | $4N^2K + 4NK^2 + N(p + 4)K$ | $3NK$ | $O(N^2K)$ |

Due to the class-driven constraint, our SGCF method needs NK more operation for addition and multiplication compared with LCCF in each iteration. For the graph Laplacian term, our SGCF and LCCF methods need to compute the weight matrix S . Because of S is a sparse matrix, we only need $N(p + 3)K$ more addition and multiplication comparing to CF in each iteration. Suppose the multiplicative updates stop after t iterations, the overall computational complexity for NMF, CF, LCCF and SGCF will be $O(tMNK)$, $O(tN^2K + N^2M)$, $O(tN^2K + N^2M) + N^2p$ and $O(tN^2K + N^2M) + N^2p + NK$. Since p is usually very small (around five) [19] and $K \ll \min\{M, N\}$, SGCF, LCCF and CF will have the same computational complexity by using the big O notation when dealing with the high-dimensional data.

4. Convergence proof

We have the following theorem regarding the above updated rules, which guarantees the SGCF method converges to a local minimum.

Theorem 4.1. The objective function O in Eq (3.1) is non-increasing under the updating rules in Eqs (3.4) and (3.5). The objective function is invariant under these updating rules if and only if \mathbf{W} and \mathbf{V} are at a stationary point.

In order to prove Theorem 4.1, we require to prove that the objective function O of SGCF is non-increasing under the updated rules in Eqs (3.4) and (3.5). Due to the fact that the graph regularization and the class-driven constraint in the objective function O of SGCF are only related to the coefficient matrix \mathbf{V} , and the iterative updated rule for Eq (3.4) is exactly the same as the updated rule for \mathbf{W} in the CDCF [41] and CF [18], we omit the convergence proof of SGCF to show that the objective function O of the optimization problem Eq (3.1) is non-increasing under the updated rule in Eq (3.4) [18, 41]. Therefore, we only have to prove that the objective function O in Eq (3.1) is non-increasing under the updated rule in Eq (3.5). Similar to the literature [28, 37], we utilize a similar auxiliary function that has been used in the expectation maximization algorithm [42]. We begin by introducing a definition and some lemmas.

Definition 4.1. $G(x, x')$ is the auxiliary function of $F(x)$ if the following conditions:

$$G(x, x') \geq F(x), \quad G(x, x) = F(x) \quad (4.1)$$

are satisfied.

Lemma 4.1. If $G(x, x')$ is the auxiliary function of $F(x)$, then $F(x)$ is non-increasing under the update

$$x^{(t+1)} = \arg \min_x G(x, x^{(t)}). \quad (4.2)$$

Proof. $F(x^{(t+1)}) \leq G(x^{(t+1)}, x^{(t)}) \leq G(x^{(t)}, x^{(t)}) = F(x^{(t)})$.

Now, we prove the convergence of the iterative updating rules for \mathbf{V} in Eq (3.5). First, we suppose that v_{ab} is any entry in the matrices \mathbf{V} , and $F_{v_{ab}}$ denotes the part of O only relevant to v_{ab} . Therefore, we have the derivative of F_{ab} with respect to v_{ab} as follows:

$$F'_{v_{ab}} = (-2\mathbf{KW} + 2\mathbf{VW}^T\mathbf{KW} + 2\alpha\mathbf{LV} + \beta\mathbf{D})_{ab}, \quad (4.3)$$

$$F''_{v_{ab}} = 2(\mathbf{W}^T\mathbf{KW})_{bb} + 2\alpha L_{aa}. \quad (4.4)$$

From Eq (3.5), we can observe that the updated rule is element-wise. That means we only need to prove $F_{z_{ab}}$ is non-increasing under the updated rule for Eq (3.5).

Lemma 4.2. Let F' denotes the first order derivative with respect to \mathbf{V} . The function

$$G(v, v_{ab}^{(t)}) = F_{v_{ab}}(v_{ab}^{(t)}) + F'_{v_{ab}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{(\mathbf{VW}^T\mathbf{KW} + \alpha\mathbf{HV} + \beta\mathbf{D}/2)_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \quad (4.5)$$

is an auxiliary function of $F_{z_{ab}}$.

Proof. Evidently, $G(v, v) = F_{v_{ab}}(v)$, we only have to show that $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$. To achieve this goal, we can compare the Taylor series expansion of $F_{v_{ab}}(v)$ at $v_{ab}^{(t)}$ with the auxiliary function $G(v, v_{ab}^{(t)})$:

$$F_{v_{ab}}(v) = F_{v_{ab}}(v_{ab}^{(t)}) + F'_{v_{ab}}(v - v_{ab}^{(t)}) + \frac{1}{2}F''_{v_{ab}}(v - v_{ab}^{(t)})^2. \quad (4.6)$$

Putting (4.4) to (4.6) and comparing with (4.5) to find that $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$ is equivalent to

$$(\mathbf{VW}^T\mathbf{KW} + \alpha\mathbf{HV} + \beta\mathbf{D}/2)_{ab} \geq \frac{1}{2}v_{ab}^{(t)}F''_{v_{ab}} = v_{ab}^{(t)}[(\mathbf{W}^T\mathbf{KW})_{bb} + \alpha L_{aa}]. \quad (4.7)$$

In order to prove the above inequality, we have:

$$(\mathbf{VW}^T\mathbf{KW})_{ab} = \sum_{l=1}^k v_{al}^t (\mathbf{W}^T\mathbf{KW})_{lb} \geq v_{ab}^{(t)} (\mathbf{W}^T\mathbf{KW})_{bb}, \quad (4.8)$$

$$\alpha(\mathbf{HV})_{ab} = \alpha \sum_{l=1}^m \mathbf{H}_{al} v_{lb}^t \geq \alpha \mathbf{H}_{aa} v_{ab}^t \geq \alpha(\mathbf{H} - \mathbf{S})_{aa} v_{ab}^t = \alpha \mathbf{L}_{aa} v_{ab}^t, \quad (4.9)$$

and $(\beta\mathbf{D}/2)_{ab} \geq 0$. Therefore, Eq (4.7) holds and we have $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$.

Now, we can show the convergence of Theorem 4.1 for \mathbf{V} :

Proof of Theorem 4.1. For the updated rule for Eq (3.5) of \mathbf{V} , we can replace $G(v, v_{ab}^{(t)})$ in Eq (4.2) by Eq (4.5) to obtain the updated rule, which is exactly the same with the iterative updated rule for \mathbf{V} .

$$\begin{aligned} v_{ab}^{(t+1)} &= \arg \min_v G(v, v_{ab}^{(t)}) \\ &= v_{ab}^{(t)} - \frac{v_{ab}^{(t)}}{2(\mathbf{VW}^T\mathbf{KW} + \alpha\mathbf{HV} + \beta\mathbf{D}/2)_{ab}} \cdot F'_{z_{ab}} \\ &= v_{ab}^{(t)} \frac{(\mathbf{KW} + \alpha\mathbf{SV})_{ab}}{(\mathbf{VW}^T\mathbf{KW} + \alpha\mathbf{HV} + (\beta\mathbf{D})/2)_{ab}}. \end{aligned} \quad (4.10)$$

Since Eq (4.5) is an auxiliary function, $F_{v_{ab}}$ is non-increasing under the updated rule for Eq (3.5) with Lemma 4.2.

5. Experimental results

In this section, we report several experiments from the performance of the proposed SGCF and compare them with the existing methods on the COIL20, PIE, Yale B and MNIST datasets. The important statistic properties of these datasets are enumerated in Table 2. We will describe the details of these datasets individually later on. The compared methods as follows:

- Traditional K-means clustering algorithm (K-means);
- Nonnegative matrix factorization (NMF) [6, 7];
- Concept factorization (CF) [18];
- Locally consistent concept factorization (LCCF) [19];
- Constrained concept factorization (CCF) [33];
- Constrained nonnegative matrix factorization (CNMF) [14];
- Discriminative nonnegative matrix factorization (DNMF) [27];
- Class-driven concept factorization (CDCF) [41];

Table 2. Properties of the datasets.

| Dataset | size (N) | dimensionality (M) | of classes |
|---------------|----------|--------------------|------------|
| <i>COIL20</i> | 1440 | 1024 | 20 |
| <i>PIE</i> | 11554 | 1024 | 68 |
| <i>YaleB</i> | 2414 | 1024 | 38 |
| <i>MNIST</i> | 70000 | 784 | 10 |

Among them, K-means, NMF, CF and LCCF are unsupervised learning algorithms, while CCF, CDCF, CNMF, DNMF and our proposed SGCF are semi-supervised learning algorithms, and K-means is conduct on original datasets. For the semi-supervised learning algorithms, we randomly picked 20% samples from each class as the available label information in each trial, and used them to construct the label indicator matrix D . In order to achieve better representative results, we always set the new dimension k of the data space equal to the cluster number P by selecting a random subset of all classes. Then K-means was repeated 20 times in the new representation \mathbf{V} and the best result was recorded. The number of nearest neighbors p was set to be five in the graph regularization methods as suggest in [34, 37], such as LCCF, CNMF, DNMF and SGCF. Two widely-used evaluation metrics, clustering accuracy (AC) and normalized mutual information (NMI), were used to evaluate the clustering performance, and detailed definitions of these evaluation metrics can be found in [14, 28, 37]. For each dateset, we made twenty independent trials and recorded the average clustering result.

5.1. Experiments on COIL20 dataset

The COIL20 dataset consists of the gray-scale images of 20 objects, and each image is manually cropped to 32×32 . These objects were placed on a motorized turntable, which was rotated 360 degrees to various object poses with respect to a fixed camera. Figure 1 shows some images from the COIL20 dataset.

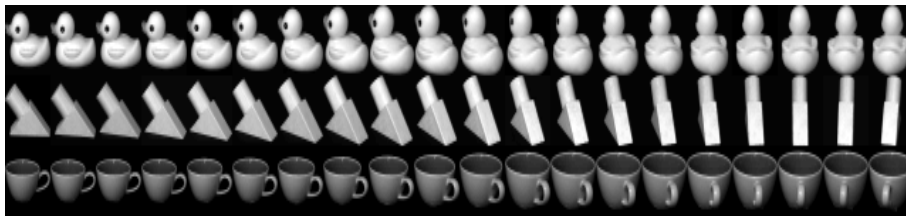


Figure 1. Sample images from the COIL20 image datasets.

Form Table 3, we can find that proposed SGCF achieves the best average clustering performance. Compared with NMF, CF and LCCF, the SGCF can achieve 15.35%, 17.41% and 3.53% improvement in AC on average, respectively. The corresponding NMI on average increases are 22.22%, 22.87% and 4.81% improvement, respectively. Therefore, the performances of SGCF surpasses those of NMF, CF and LCCF. This is mainly due to the fact that NMF, CF and LCCF are unsupervised learning algorithms, while SGCF is a semi-supervised learning algorithm, which considers the label information by introducing the class-driven constraint.

Table 3. Clustering performance on the COIL20 dataset.

| P | Accuracy (%) | | | | | | | | | NMI (%) | | | | | | | | |
|------|--------------|-------|-------|---------------|-------|-------|-------|-------|---------------|---------|-------|-------|---------------|-------|-------|-------|-------|---------------|
| | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF |
| 2 | 91.18 | 88.65 | 91.39 | 100.00 | 91.77 | 93.09 | 90.10 | 91.18 | 100.00 | 75.33 | 64.34 | 73.20 | 100.00 | 73.23 | 74.81 | 70.39 | 70.47 | 100.00 |
| 3 | 87.15 | 84.44 | 83.54 | 96.06 | 89.54 | 91.87 | 88.56 | 89.40 | 99.33 | 75.68 | 70.24 | 71.53 | 92.87 | 78.87 | 80.89 | 76.61 | 76.62 | 98.29 |
| 4 | 80.02 | 75.73 | 76.32 | 89.67 | 80.38 | 85.73 | 78.16 | 84.27 | 94.06 | 72.48 | 67.79 | 68.31 | 87.43 | 71.45 | 75.82 | 71.53 | 74.91 | 91.23 |
| 5 | 75.82 | 71.58 | 71.51 | 85.22 | 76.21 | 83.24 | 76.50 | 79.51 | 92.01 | 70.28 | 65.39 | 66.29 | 83.27 | 68.20 | 74.02 | 69.94 | 71.30 | 89.83 |
| 6 | 77.19 | 72.30 | 69.97 | 87.75 | 70.32 | 84.84 | 79.64 | 81.54 | 92.16 | 73.30 | 70.08 | 67.99 | 86.20 | 68.42 | 78.66 | 76.36 | 76.31 | 91.34 |
| 7 | 76.15 | 74.88 | 71.26 | 86.99 | 73.58 | 85.32 | 82.30 | 82.64 | 89.02 | 75.53 | 74.89 | 70.59 | 87.64 | 72.34 | 80.73 | 79.65 | 78.95 | 90.82 |
| 8 | 72.52 | 72.11 | 68.61 | 82.30 | 71.02 | 82.59 | 77.21 | 79.22 | 88.39 | 73.31 | 71.57 | 70.13 | 84.08 | 71.26 | 78.36 | 75.19 | 76.31 | 90.20 |
| 9 | 75.19 | 72.45 | 66.33 | 81.02 | 69.68 | 82.45 | 79.19 | 78.98 | 88.47 | 76.13 | 73.40 | 68.66 | 83.68 | 72.03 | 79.88 | 78.46 | 78.36 | 90.75 |
| 10 | 70.82 | 69.94 | 64.24 | 79.08 | 67.52 | 81.06 | 76.24 | 77.83 | 83.68 | 74.03 | 72.62 | 67.83 | 81.84 | 70.10 | 79.27 | 76.98 | 77.63 | 87.89 |
| Avg. | 78.54 | 75.75 | 73.69 | 87.57 | 76.67 | 85.58 | 80.88 | 82.73 | 91.10 | 74.01 | 70.04 | 69.39 | 87.45 | 71.76 | 78.05 | 75.01 | 75.65 | 92.26 |

5.2. Experiments on PIE face dataset

The PIE dataset contains 11,554 face images of 68 different people, each of which has 42 facial images under different light and illumination conditions. Each image is manually cropped to 32×32 . For the fixed cluster number P , we randomly selected 100 images of each class in this experiment. Figure 2 shows some images from the PIE face dataset.

Table 4 shows that our proposed SGCF achieves the best performance for all different cluster number P . Specifically, the AC and NMI of LCCF are 32.57% and 12.35% with 5.00% and 8.88% gains over CF, respectively. The AC and NMI of SGCF are 62.08% and 44.74% with gains of 29.51% and 32.39% over LCCF, respectively. Therefore, the performance of SGCF is superior to those of LCCF and CF. The main reason is that CF and LCCF are unsupervised learning methods, while SGCF integrates the label information into CF by utilizing the class-driven constraint. Meanwhile, it is noteworthy to point out that SGCF obtains the best performance due to the fact that SGCF effectively takes both the class-driven constraint and graph regularization into account.



Figure 2. Sample face images from the PIE face dataset.

Table 4. Clustering results comparison on the PIE face dataset.

| P | Accuracy (%) | | | | | | | | | NMI (%) | | | | | | | | |
|------|--------------|-------|-------|-------|-------|-------|-------|-------|--------------|---------|-------|------|-------|-------|-------|-------|-------|--------------|
| | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF |
| 2 | 51.73 | 51.78 | 52.32 | 54.73 | 52.07 | 59.33 | 52.57 | 62.09 | 76.78 | 0.16 | 0.15 | 0.20 | 4.02 | 0.27 | 2.79 | 0.61 | 4.65 | 40.20 |
| 3 | 38.80 | 42.07 | 40.78 | 44.53 | 46.37 | 51.20 | 45.81 | 54.45 | 74.17 | 3.77 | 5.63 | 4.00 | 8.88 | 9.18 | 11.27 | 11.97 | 14.55 | 51.11 |
| 4 | 29.81 | 32.59 | 31.11 | 34.81 | 36.50 | 45.24 | 40.73 | 48.83 | 64.72 | 2.29 | 5.59 | 2.87 | 8.36 | 8.14 | 12.00 | 15.23 | 16.66 | 43.58 |
| 5 | 26.29 | 31.89 | 27.57 | 32.92 | 33.02 | 40.69 | 34.75 | 42.92 | 64.41 | 3.91 | 10.87 | 4.39 | 13.99 | 10.50 | 15.42 | 15.33 | 17.54 | 47.32 |
| 6 | 22.59 | 27.89 | 23.91 | 28.43 | 28.79 | 36.33 | 34.49 | 41.83 | 60.68 | 3.64 | 11.75 | 4.70 | 13.81 | 9.09 | 14.45 | 20.08 | 20.44 | 45.59 |
| 7 | 20.18 | 27.12 | 20.51 | 27.00 | 25.54 | 32.63 | 34.30 | 39.89 | 59.16 | 3.35 | 14.82 | 3.54 | 14.63 | 8.50 | 13.31 | 24.25 | 24.29 | 45.23 |
| 8 | 19.67 | 26.81 | 19.15 | 25.53 | 25.08 | 31.92 | 33.90 | 37.72 | 55.05 | 6.81 | 18.05 | 4.49 | 16.13 | 10.82 | 16.04 | 25.61 | 26.41 | 43.61 |
| 9 | 18.35 | 25.65 | 16.88 | 23.16 | 21.58 | 28.44 | 30.88 | 34.95 | 53.72 | 7.21 | 19.37 | 3.35 | 15.50 | 8.31 | 14.60 | 26.02 | 26.81 | 44.06 |
| 10 | 17.74 | 24.58 | 15.94 | 22.03 | 20.18 | 26.55 | 30.04 | 32.93 | 50.04 | 8.59 | 20.92 | 3.65 | 15.86 | 8.53 | 14.51 | 26.78 | 26.68 | 41.96 |
| Avg. | 27.24 | 32.26 | 27.57 | 32.57 | 32.13 | 39.15 | 37.49 | 43.96 | 62.08 | 4.41 | 11.91 | 3.47 | 12.35 | 8.15 | 12.71 | 18.43 | 19.77 | 44.74 |

5.3. Experiments on Yale B dataset

The Extended Yale B dataset contains 2,414 frontal face images from 38 objects, each of which contains 64 face images captured under controlled lighting in the laboratory. Each image is manually cropped to 32×32 . In this experiment, we randomly selected 50 images of each class for the fixed cluster number P . Figure 3 shows some images from the Yale B dataset.



Figure 3. Sample face images from the Yale B face dataset.

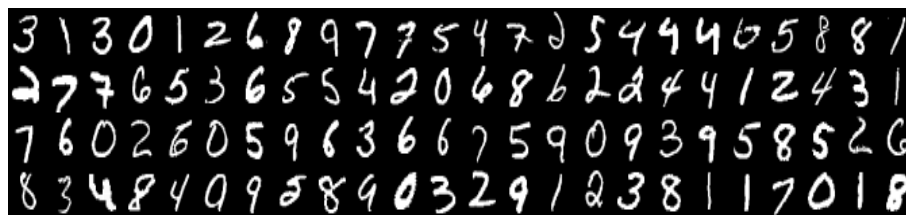
From Table 5, it can be observed that SGCF achieves the best performance in clustering. The average clustering accuracies obtained by k-means, NMF, CF, LCCF, CCF, CDCF, CNMF, DNMF and SGCF are 25.67%, 28.14%, 26.67%, 49.10%, 30.35%, 39.82%, 32.59%, 35.21% and 65.42%, respectively. Compared with DNMF, SGCF can achieve 30.21% improvement in AC and 39.70% improvement in NMI. The important reason is that DNMF cannot capture the intrinsic geometrical structure of data space, while SGCF can efficiently detect the intrinsic structure of data space by using the graph regularization. Thus, the clustering performance can be greatly improved.

Table 5. Clustering results comparison on the Yale B face dataset.

| P | Accuracy (%) | | | | | | | | NMI (%) | | | | | | | | | |
|------|--------------|-------|-------|-------|-------|-------|-------|-------|--------------|--------|-------|------|-------|------|-------|-------|-------|--------------|
| | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF |
| 2 | 51.95 | 52.35 | 52.25 | 69.25 | 53.15 | 60.15 | 53.85 | 59.45 | 89.60 | 0.20 | 0.22 | 0.23 | 32.35 | 0.57 | 3.29 | 0.73 | 8.05 | 65.36 |
| 3 | 35.97 | 36.30 | 36.77 | 54.40 | 40.33 | 48.27 | 39.53 | 45.03 | 75.07 | 0.42 | 0.53 | 0.66 | 27.42 | 2.39 | 5.99 | 1.94 | 11.98 | 50.50 |
| 4 | 28.35 | 28.88 | 29.87 | 50.62 | 34.40 | 42.90 | 30.43 | 37.88 | 70.02 | 0.74 | 0.97 | 1.70 | 34.62 | 4.10 | 9.57 | 2.00 | 13.63 | 53.31 |
| 5 | 24.08 | 24.34 | 26.46 | 48.40 | 29.34 | 40.46 | 29.44 | 33.36 | 66.16 | 1.63 | 1.66 | 3.40 | 39.15 | 4.99 | 13.03 | 7.18 | 13.87 | 54.50 |
| 6 | 19.87 | 22.93 | 22.40 | 45.12 | 26.78 | 37.70 | 28.00 | 31.93 | 61.85 | 0.81 | 4.40 | 2.65 | 37.71 | 6.24 | 13.30 | 10.56 | 16.81 | 52.22 |
| 7 | 18.24 | 22.44 | 20.16 | 45.51 | 24.93 | 35.37 | 28.59 | 29.01 | 60.89 | 2.49 | 7.97 | 2.87 | 41.21 | 7.42 | 15.10 | 14.95 | 16.15 | 53.30 |
| 8 | 18.28 | 23.06 | 19.12 | 45.65 | 23.31 | 33.66 | 28.61 | 27.80 | 56.89 | 4.08 | 11.37 | 3.83 | 42.35 | 7.96 | 16.34 | 18.44 | 16.69 | 53.29 |
| 9 | 17.73 | 21.37 | 16.96 | 42.12 | 20.82 | 30.30 | 27.00 | 27.11 | 54.97 | 5.23 | 11.45 | 3.47 | 42.89 | 7.35 | 14.21 | 19.71 | 17.14 | 52.84 |
| 10 | 16.59 | 21.59 | 16.03 | 40.87 | 20.13 | 29.54 | 27.84 | 25.31 | 53.32 | 5.60 | 13.07 | 3.89 | 41.28 | 8.10 | 15.66 | 22.03 | 16.86 | 50.18 |
| Avg. | 25.67 | 28.14 | 26.67 | 49.10 | 30.35 | 39.82 | 32.59 | 35.21 | 65.42 | 2.36 | 5.74 | 2.52 | 37.66 | 5.46 | 11.83 | 10.84 | 14.24 | 53.94 |

5.4. Experiments on MNIST database

The MNIST dataset classifies 10 classes from zero to nine, which provides a total of 70,000 handwritten samples. Each image is resized to 28×28 gray-scale images. Similar to experiments in [17], we randomly choose 100 images of each class in this experiment. Figure 4 shows some handwritten samples from the MNIST dataset.

**Figure 4.** Sample face images from the MNIST databases.**Table 6.** Clustering performance on the MNIST dataset.

| P | Accuracy (%) | | | | | | | | NMI (%) | | | | | | | | | |
|------|--------------|-------|-------|-------|-------|-------|-------|-------|--------------|--------|-------|-------|-------|-------|-------|-------|-------|--------------|
| | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF | kmeans | NMF | CF | LCCF | CCF | CDCF | CNMF | DNMF | SGCF |
| 2 | 92.03 | 89.35 | 92.23 | 93.78 | 94.00 | 93.65 | 89.38 | 89.38 | 96.80 | 67.81 | 59.87 | 67.74 | 74.24 | 72.29 | 71.27 | 57.97 | 57.94 | 83.42 |
| 3 | 72.75 | 70.67 | 73.55 | 73.83 | 80.47 | 86.87 | 79.82 | 83.58 | 92.37 | 52.63 | 48.37 | 49.64 | 53.91 | 59.94 | 64.07 | 59.07 | 60.81 | 75.89 |
| 4 | 65.88 | 61.72 | 64.89 | 68.24 | 69.79 | 82.03 | 68.01 | 69.96 | 87.84 | 53.10 | 45.15 | 47.60 | 53.57 | 56.24 | 61.47 | 53.41 | 54.21 | 70.19 |
| 5 | 63.36 | 58.65 | 59.41 | 62.40 | 66.28 | 78.07 | 69.04 | 71.88 | 84.42 | 49.60 | 43.84 | 44.58 | 49.73 | 53.38 | 58.62 | 53.85 | 53.75 | 67.01 |
| 6 | 59.23 | 52.56 | 54.00 | 54.93 | 63.12 | 78.17 | 61.91 | 66.89 | 83.47 | 51.40 | 45.10 | 45.49 | 49.23 | 54.03 | 61.35 | 54.45 | 53.59 | 68.07 |
| 7 | 58.86 | 53.13 | 53.16 | 54.11 | 60.72 | 75.69 | 63.28 | 67.81 | 80.23 | 50.57 | 44.61 | 43.74 | 48.25 | 53.18 | 59.61 | 54.19 | 53.86 | 65.60 |
| 8 | 54.50 | 49.01 | 49.16 | 51.57 | 57.26 | 72.47 | 59.63 | 64.43 | 76.71 | 49.26 | 43.09 | 42.87 | 48.07 | 50.03 | 57.94 | 52.14 | 52.35 | 62.65 |
| 9 | 52.94 | 47.64 | 49.01 | 51.53 | 57.72 | 71.09 | 59.54 | 63.13 | 76.46 | 48.69 | 42.97 | 44.03 | 48.28 | 51.29 | 57.43 | 52.52 | 52.95 | 63.16 |
| 10 | 51.32 | 46.84 | 48.37 | 48.89 | 55.50 | 69.25 | 56.62 | 62.81 | 74.78 | 48.83 | 44.04 | 44.43 | 47.60 | 50.34 | 56.87 | 52.34 | 52.88 | 62.24 |
| Avg. | 63.43 | 58.84 | 60.42 | 62.14 | 67.21 | 78.59 | 67.47 | 71.09 | 83.68 | 52.43 | 46.34 | 47.79 | 52.54 | 55.64 | 60.96 | 54.44 | 54.70 | 68.69 |

We can see from Table 6 that the proposed SGCF obtains the best performance compared with the others methods. Specifically, compared with CCF and CDCF, SGCF can achieve 16.47% and 5.09% improvement in AC, respectively. For NMI, SGCF can achieve 13.05% and 7.73% improvement, respectively. Therefore, the performance of SGCF is significantly superior to the CCF and CDCF. The

major reason is that CCF and CDCF do not consider the graph regularization, while SGCF not only preserves the intrinsic structure of data space by using the graph regularization, but also considers the available label information by utilizing the class-driven constraints. Besides, we can see that the semi-supervised learning methods (i.e., CCF, CDCF, CNMF, DNMF and SGCF) consistently outperform other unsupervised learning methods, such as K-means, NMF, CF and LCCF. This indicates that the limited label information can improve the clustering performance for semi-supervised learning methods, which proves the effectiveness of SGCF yet again.

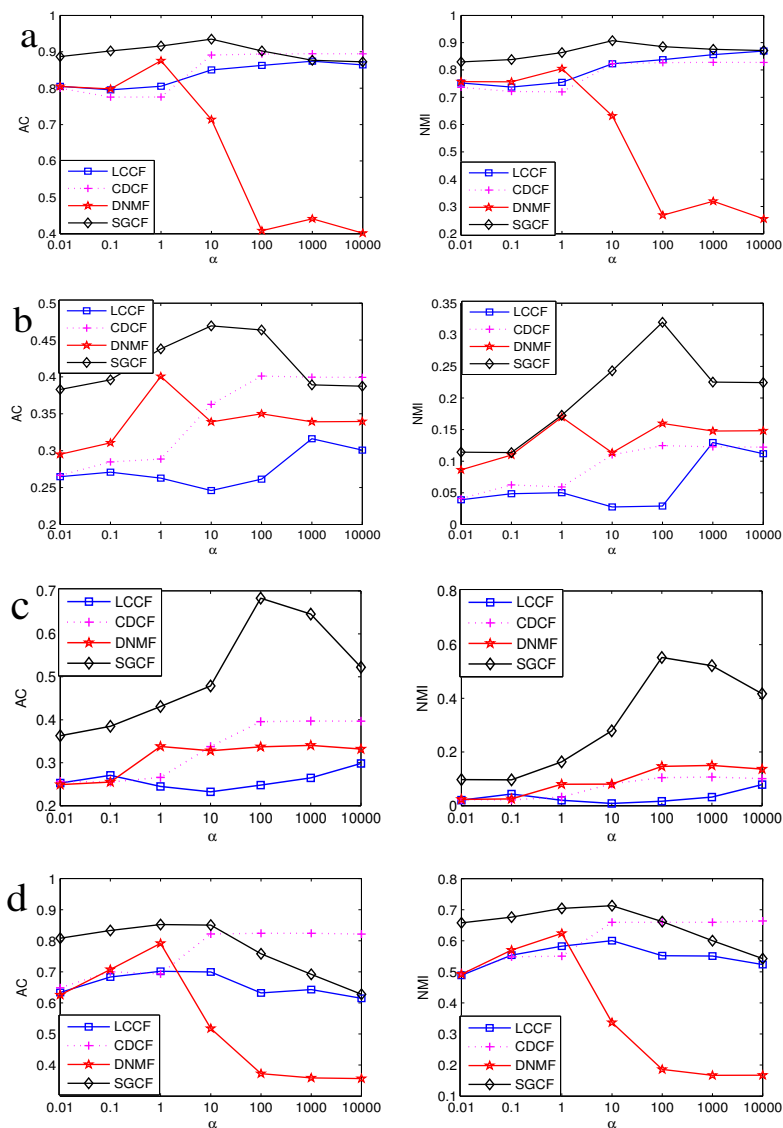


Figure 5. The performance with the varied parameter α . a: COIL20, b: PIE, c: YaleB and d: MNIST.

5.5. Parameters analysis

In this subsection, we show how the tradeoff parameters α, β can influence the performance of clustering. Here, we randomly choose 50 images of each class and randomly select five classes from each dataset as a subset of dataset in these experiments. Following several existing works [23, 27, 32, 37], we separately analyze α and β in SGCF. First, we empirically fix the parameters β , and then search the optimal parameters α from [0.01, 0.1, 1, 10, 100, 1000]. Second, we fix the optimal parameters α , and search the optimal parameters β from a range of [0.01, 0.1, 1, 10, 100, 1000, 10000] in SGCF. We independently run on 20 times for each experiment, and the results are shown in Figures 5 and 6. The parameter setting of LCCF, CDCF, DNMF and SGCF are shown in Table 7.

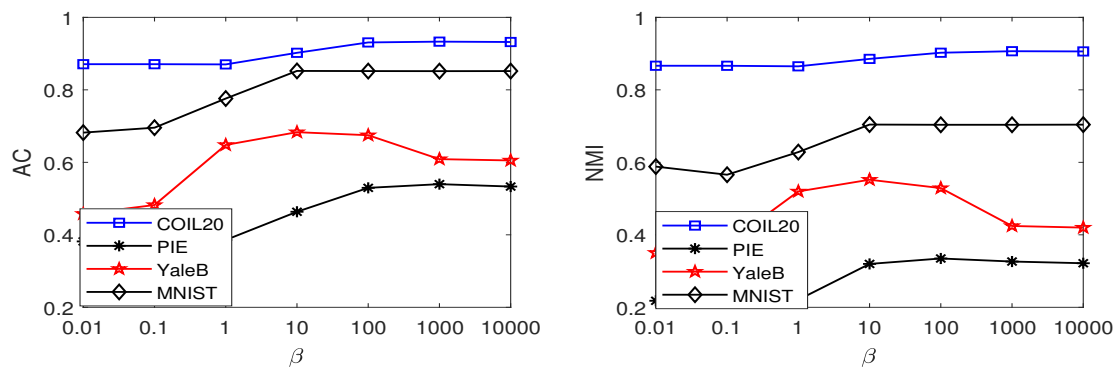


Figure 6. The performance of SGCF with the varied parameter β .

Table 7. Parameter setting of LCCF, CDCF and SGCF.

| Dataset | LCCF | CDCF | DNMF | SGCF |
|---------------|------------------|------------------|-----------------|-----------------------------|
| <i>COIL20</i> | $\alpha = 1000$ | $\alpha = 1000$ | $\alpha = 1$ | $\alpha = 10, \beta = 1000$ |
| <i>PIE</i> | $\alpha = 1000$ | $\alpha = 100$ | $\alpha = 1$ | $\alpha = 10, \beta = 1000$ |
| <i>YaleB</i> | $\alpha = 10000$ | $\alpha = 10000$ | $\alpha = 1000$ | $\alpha = 100, \beta = 10$ |
| <i>MNIST</i> | $\alpha = 1$ | $\alpha = 100$ | $\alpha = 1$ | $\alpha = 10, \beta = 10$ |

It's worth noting that the proposed SGCF is a semi-supervised learning method, and the performance of the algorithm is closely related to the proportion of labeled data. In order to demonstrate how the performance of SGCF varies with labeled data, we select the proportions of labeled data from each class with the label percentage varying from 10% to 50% as the labeled data constructs the label indicator matrix D for each dataset. We randomly choose five classes from each dataset and independently run on 20 times for each experiment, and the experimental results are shown in Figure 7. From Figure 7, we can see that the clustering performance of CCF, CDCF, CNMF, DNMF and SGCF are generally improved with the increase of the proportion of the labeled data, while the unsupervised learning methods K-means, NMF, CF and LCCF have clustering performances that are not changed because it's irrespective of the proportion of the labeled data variation. However, it can be found that the SGCF can achieve better performance when the proportion of the labeled data is small. Therefore, the proposed SGCF is a more robust method.

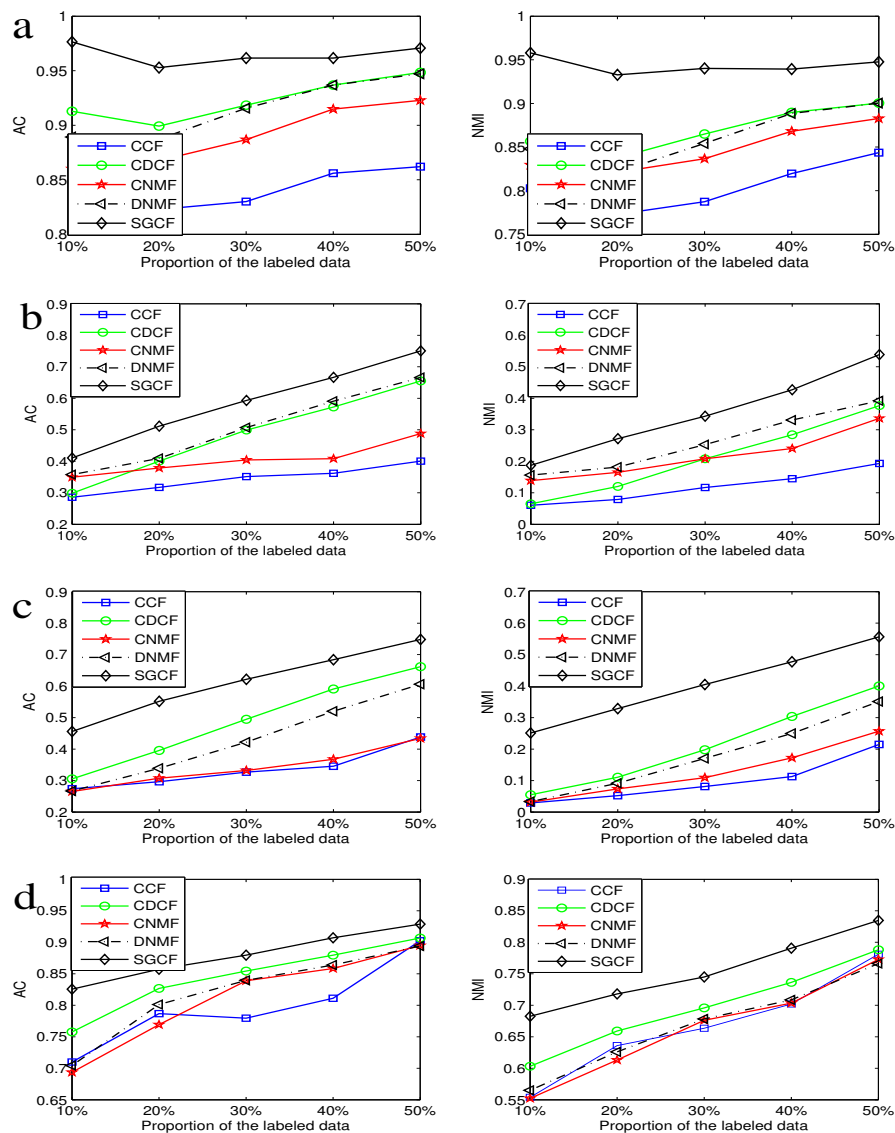


Figure 7. The performance with a varied proportion of the labeled data. a: COIL20, b: PIE, c: YaleB and d: MNIST.

6. Conclusions

In this paper, a new SGCF with the class-driven constraint was proposed. In this framework, the label information with the class-driven constraint was encoded in CF, which forces the new representations of samples to be more similar within the same class while different between unlike classes. To extract the geometric information of the original data space, a graph regularization was added to the objective function of SGCF. For the proposed SGCF method, iterative updated rules were proposed and the convergence was proven both theoretically and experimentally. The experimental results on four databases have demonstrated the better performance of SGCF in comparison to the other state-of-the-art matrix factorization methods.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Key Project of Ningxia Natural Science Foundation under Grant no. 2022AAC02043, the First-class Discipline Construction Fund project of Ningxia Higher Education under Grant no. NXYLXK2017B09, the Natural Science Foundation of Shaanxi Province under Grant no. 2020JM-630, the Major scientific Research Project of Northern University for Nationalities under Grant no. ZDZX201901, the Shangluo Universities Key Disciplines Project (Discipline name: Mathematics) and the Fundamental Research Fund of Shangluo University under Grant no. 18SKY009, 19SCX02.

Conflict of interest

The authors declare that they have no conflicts of interest related to this article.

References

1. H. P. Kriegel, P. Kroger, A. Zimek, Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering, *ACM T. Knowl. Discov. D.*, **3** (2009), 1–58. <http://doi.org/10.1145/1497577.1497578>
2. G. Cui, Y. Li, Nonredundancy regularization based nonnegative matrix factorization with manifold learning for multiview data representation, *Inform. Fusion*, **82** (2022), 86–98. <http://doi.org/10.1016/j.inffus.2021.12.001>
3. J. A. Lee, M. Verleysen, *Nonlinear dimensionality reduction*, New York: Springer, 2007. <https://doi.org/10.1007/978-0-387-39351-3>
4. I. T. Jolliffe, *Principal component analysis*, New York: Springer, 1986. <http://doi.org/10.1007/978-1-4757-1904-8>
5. D. Kalman, A singularly valuable decomposition: The SVD of a matrix, *Coll. Math. J.*, **27** (1996), 2–23. <https://doi.org/10.1080/07468342.1996.11973744>
6. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401** (1999), 788–791. <https://doi.org/10.1038/44565>
7. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, *NIPS'00: Proceedings of the 13th international conference on neural information processing systems*, 2000, 535–541.
8. G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE T. Pattern Anal. Mach. Intell.*, **35** (2013), 171–184. <https://doi.org/10.1109/TPAMI.2012.88>
9. J. Liu, Y. Chen, J. Zhang, Z. Xu, Enhancing low-rank subspace clustering by manifold regularization, *IEEE T. Image Process.*, **23** (2014), 4022–4030. <https://doi.org/10.1109/TIP.2014.2343458>

10. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, *J. Am. Soc. Inform. Sci.*, **41** (1990), 391–407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-ASII>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASII>3.0.CO;2-9)
11. W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, *SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information*, 2003, 267–273. <https://doi.org/10.1145/860435.860485>
12. F. Shahnaz, M. W. Berry, V. P. Pauca, R. J. Plemmons, Document clustering using nonnegative matrix factorization, *Inform. Process. Manag.*, **42** (2006), 373–386. <https://doi.org/10.1016/j.ipm.2004.11.005>
13. X. Long, H. Lu, Y. Peng, W. Li, Graph regularized discriminative non-negative matrix factorization for face recognition, *Multimed Tools Appl.*, **72** (2014), 2679–2699. <https://doi.org/10.1007/s11042-013-1572-z>
14. H. Liu, Z. Wu, X. Li, D. Cai, T. S. Huang, Constrained nonnegative matrix factorization for image representation, *IEEE T. Pattern Anal. Mach. Intell.*, **34** (2012), 1299–1311. <https://doi.org/10.1109/TPAMI.2011.217>
15. J. P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, *PNAS*, **101** (2004), 4164–4169. <https://doi.org/10.1073/pnas.0308531101>
16. W. Hua, X. He, Discriminative concept factorization for data representation, *Neurocomputing*, **74** (2011), 3800–3807. <https://doi.org/10.1016/j.neucom.2011.07.020>
17. X. Peng, D. Chen, D. Xu, Hyperplane-based nonnegative matrix factorization with label information, *Inform. Sci.*, **493** (2019), 1–19. <https://doi.org/10.1016/j.ins.2019.04.026>
18. W. Xu, Y. Gong, Document clustering by concept factorization, *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information*, 2004, 202–209. <https://doi.org/10.1145/1008992.1009029>
19. D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, *IEEE T. Knowl. Data Eng.*, **23** (2011), 902–913. <https://doi.org/10.1109/TKDE.2010.165>
20. H. Liu, Z. Yang, J. Yang, Z. Wu, X. Li, Local coordinate concept factorization for image representation, *IEEE T. Neur. Net. Lear. Syst.*, **25** (2014), 1071–1082. <https://doi.org/10.1109/TNNLS.2013.2286093>
21. D. Wei, X. Shen, Q. Sun, X. Gao, Z. Ren, Adaptive graph guided concept factorization on Grassmann manifold, *Inform. Sci.*, **576** (2021), 725–742. <https://doi.org/10.1016/j.ins.2021.08.040>
22. S. Peng, W. Ser, B. Chen, L. Sun, Z. Lin, Correntropy based graph regularized concept factorization for clustering, *Neurocomputing*, **316** (2018), 34–48. <https://doi.org/10.1016/j.neucom.2018.07.049>
23. H. Li, J. Zhang, J. Liu, Graph-regularized CF with local coordinate for image representation, *J. Vis. Commun. Image Rep.*, **49** (2017), 392–400. <https://doi.org/10.1016/j.jvcir.2017.10.005>
24. Y. He, H. Lu, L. Huang, S. Xie, Pairwise constrained concept factorization for data representation, *Neural Networks*, **52** (2014), 1–17. <https://doi.org/10.1016/j.neunet.2013.12.007>
25. L. Xue, S. Xiaobo, S. Zhenqiu, Y. Qiaolin, Z. Chunxia, Graph regularized multilayer concept factorization for data representation, *Neurocomputing*, **238** (2017), 139–151. <https://doi.org/10.1016/j.neucom.2017.01.045>

26. H. Cai, B. Liu, Y. Xiao, L. Y. Lin, Semi-supervised multi-view clustering based on constrained nonnegative matrix factorization, *Knowl. Based Syst.*, **182** (2019), 104798. <https://doi.org/10.1016/j.knosys.2019.06.006>
27. M. Babae, S. Tsoukalas, M. Babae, G. Rigoll, M. Datcu, Discriminative nonnegative matrix factorization for dimensionality reduction, *Neurocomputing*, **173** (2016), 212–223. <https://doi.org/10.1016/j.neucom.2014.12.124>
28. S. Peng, W. Ser, B. Chen, Z. Lin, Robust semi-supervised nonnegative matrix factorization for image clustering, *Pattern Recogn.*, **111** (2021), 107683. <https://doi.org/10.1016/j.patcog.2020.107683>
29. H.g Li, J. Zhang, G. Shi, J. Liu, Graph-based discriminative nonnegative matrix factorization with label information, *Neurocomputing*, **266** (2017), 91–100. <https://doi.org/10.1016/j.neucom.2017.04.067>
30. Y. Yi, Y. Chen, J. Wang, G. Lei, J. Dai, H. Zhang, Joint feature representation and classification via adaptive graph semi-supervised nonnegative matrix factorization, *Signal Process.-Image*, **89** (2020), 115984. <https://doi.org/10.1016/j.image.2020.115984>
31. Z. Shu, C. Zhao, P. Huang, Local regularization concept factorization and its semi-supervised extension for image representation, *Neurocomputing*, **158** (2015), 1–12. <https://doi.org/10.1016/j.neucom.2015.02.014>
32. Z. Xing, Y. Ma, X. Yang, F. Nie, Graph regularized nonnegative matrix factorization with label discrimination for data clustering, *Neurocomputing*, **440** (2021), 297–309. <https://doi.org/10.1016/j.neucom.2021.01.064>
33. H. Liu, G. Yang, Z. Wu, D. Cai, Constrained concept factorization for image representation, *IEEE T. Cybernetics*, **44** (2014), 1214–1224. <https://doi.org/10.1109/TCYB.2013.2287103>
34. H. Li, Y. Gao, J. Liu, J. Zhang, C. Li, Semi-supervised graph regularized nonnegative matrix factorization with local coordinate for image representation, *Signal Process.-Image*, **102** (2022), 116589. <https://doi.org/10.1016/j.image.2021.116589>
35. S. Peng, Z. Yang, F. Nie, B. Chen, Z. Lin, Correntropy based semi-supervised concept factorization with adaptive neighbors for clustering, *Neural Networks*, **154** (2022), 203–217. <https://doi.org/10.1016/j.neunet.2022.07.021>
36. W. Yan, B. Zhang, S. Ma, Z. Yang, A novel regularized concept factorization for document clustering, *Knowl.-Based Syst.*, **135** (2017), 147–158. <https://doi.org/10.1016/j.knosys.2017.08.010>
37. D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE T. Pattern Anal.*, **33** (2011), 1548–1560. <https://doi.org/10.1109/TPAMI.2010.231>
38. C. Cortes, M. Mohri, On transductive regression, *NIPS'06: Proceedings of the 19th international conference on neural information processing systems*, 2006, 305–312.
39. F. R. K. Chung, Spectral graph theory, *American Mathematical Society*, 1997.

40. Y. H. Xiao, Z. f. Zhu, Y. Zhao, Y. C. Wei, Class-driven non-negative matrix factorization for image representation, *J. Comput. Sci. Technol.*, **28** (2013), 751–761. <https://doi.org/10.1007/s11390-013-1374-9>
41. H. Li, J. Zhang, J. Liu, Class-driven concept factorization for image representation, *Neurocomputing*, **190** (2016), 197–208. <https://doi.org/10.1016/j.neucom.2016.01.017>
42. A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the *EM* algorithm, *J. Royal Stat. Soc. Ser. B*, **39** (1977), 1–22. <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)