Correntropy based graph regularized concept factorization for clustering

Peng, Siyuan; Ser, Wee; Chen, Badong; Sun, Lei; Lin, Zhiping

2018

Peng, S., Ser, W., Chen, B., Sun, L., & Lin, Z. (2018). Correntropy based graph regularized concept factorization for clustering. *Neurocomputing*, 316, 34-48.
doi:10.1016/j.neucom.2018.07.049

<https://hdl.handle.net/10356/136678>

<https://doi.org/10.1016/j.neucom.2018.07.049>

© 2018 Elsevier B.V. All rights reserved. This paper was published in *Neurocomputing* and is made available with permission of Elsevier B.V.

Downloaded on 26 Apr 2023 15:32:45 SGT

Correntropy based Graph Regularized Concept Factorization for Clustering

Siyuan Peng^a, Wee Ser^a, Badong Chen^b, Lei Sun^c, Zhiping Lin^{a,*}

^a*School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore*

^b*Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049 China*

^c*School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China*

Abstract

Concept factorization (CF) technique is one of the most powerful approaches for feature learning, and has been successfully adopted in a wide range of practical applications such as data mining, computer vision, and information retrieval. Most existing concept factorization methods mainly minimize the square of the Euclidean distance, which is seriously sensitive to non-Gaussian noises or outliers in the data. To alleviate the adverse influence of this limitation, in this paper, a robust graph regularized concept factorization method, called correntropy based graph regularized concept factorization (GCCF), is proposed for clustering tasks. Specifically, based on the maximum correntropy criterion (MCC), GCCF is derived by incorporating the graph structure information into our proposed objective function. A half-quadratic optimization technique is adopted to solve the non-convex objective function of the GCCF method effectively. In addition, algorithm analysis of GCCF is studied. Extensive experiments on real world datasets outliers demonstrate that the proposed GCCF method outperforms seven competing methods for clustering applications.

Keywords: concept factorization, graph regularization, maximum correntropy

*Corresponding author

Email address: EZPLin@ntu.edu.sg (Zhiping Lin)

criterion, non-Gaussian noises, clustering.

1. Introduction

In the last decade, the dimensionality of the raw data became extremely high, ranging from several hundreds to thousands. Although high dimensional data usually contain more useful information, they not only obviously increase
5 memory and computation requirements but also may degrade the performance seriously due to redundancy and noises [1]. Consequently, low dimensional data representation techniques have attracted a lot of attention, and are widely used in various applications including data mining, computer vision, and information retrieval [2, 3, 4, 5, 6]. One commonly used low dimensional
10 data representation technique is matrix factorization [7], which aims to obtain two or more lower dimensional matrices, whose product is as close to the original data matrix as possible. Several typical matrix factorization methods are vector quantization (VQ)[8], singular value decomposition (SVD)[9], principal component analysis (PCA)[10, 11], nonnegative matrix factorization
15 (NMF)[12, 13], concept factorization (CF)[14, 15], and so on.

Among the matrix factorization methods, NMF and CF are two effective tools which have been frequently applied in data analysis tasks. The goal of NMF is to find two non-negative matrices (i.e., the basis matrix and the coefficient matrix), whose product provides a good approximation to the raw
20 data matrix [16]. Particularly, the nonnegative constraints imposed on the learned component matrices lead to a parts-based representation since the updating rules only allow additive, not subtractive operations. Due to the good property, NMF has demonstrated superior performance in document and image clustering tasks [17, 18, 19]. However, NMF also has some limitations.
25 For example, it cannot be used in any data with negative values. As a variation of NMF, CF mainly strives to overcome the shortcoming of NMF and meanwhile inheriting all the merits of NMF. Specifically, CF can be performed on any data involving positive and negative values[20, 21]. The CF

method was originally proposed in [14], which develops a cluster model where
 30 each cluster is represented by a linear combination of the data points, and
 each data point is expressed by a linear combination of the cluster centers.
 Recently, many variants have been developed for improving the original CF
 from different views. For instance, Cai *et al.* [21] exploited the intrinsic
 geometrical information of the data and incorporated it as an additional
 35 regularization term into the objective function, and proposed a locally
 consistent CF algorithm (LCCF); Ye and Jin presented a dual-graph
 regularized concept factorization (GCF), which considered the geometric
 structures of both the data manifold and the feature manifold [22]; Liu *et al.*
 imposed a locality constraint into the original CF algorithm to develop a local
 40 coordinate concept factorization (LCF)[23].

Up to now, most of the concept factorization methods adopt the Euclidean
 norm to measure the distance between the original data matrix and the
 reconstructed one, due to its properties such as mathematical tractability,
 computational simplicity and optimality under zero mean Gaussian noise.
 45 However, in many real-world applications, the data frequently involve different
 types of non-Gaussian noises (such as the occlusion and corruption problems
 in image clustering tasks) or outliers [24, 25]. In such situations, the
 performance of these CF based methods may degrade seriously. In order to
 eliminate the adverse effects of non-Gaussian noises or outliers, various robust
 50 similarity measures have been proposed. Among these approaches, the
 maximum correntropy criterion (MCC) in information theoretic learning (ITL)
 [26], has been successfully employed to a broad range of applications including
 face recognition [27], shape matching [28], clustering [29, 30], adaptive filtering
 [31, 32] and subspace learning [33, 34], due to its robustness and simplicity.
 55 Correntropy is a nonlinear and local similarity measure, which is directly
 related to the probability of how similar two random variables are in a
 neighborhood of the joint space. Different from the Euclidean norm which
 only contains the second-order statistics, correntropy can capture higher-order
 statistics [26, 32, 35]. Therefore, MCC can provide potentially significant

60 performance improvement when the data involves non-Gaussian noises or outliers. However, to the best of our knowledge, MCC based concept factorization algorithm has not been studied yet in the literature.

In this paper, we propose a robust graph regularized concept factorization method, called correntropy based graph regularized concept factorization 65 (GCCF), which is robust to non-Gaussian noises and outliers in the raw data. The main contributions of the present work are summarized as follows:

1. Instead of using the Euclidean norm, we adopt correntropy as the similarity measure to calculate the distance between the raw data matrix and the reconstructed one, and accordingly obtain the correntropy based 70 objective function. By incorporating the intrinsic graph structure information of the data as an additional regularization term into the correntropy based objective function, we derive the objective function of GCCF.
2. A half-quadratic optimization technique [33, 36] is used to convert the 75 non-convex objective function of GCCF into a quadratic function of the multiplicative form. Therefore, the optimization problem is reduced to an iteratively reweighed CF with a regularization term, which can be solved by a multiplicative update method.
3. The convergence of the proposed GCCF method is proven theoretically, 80 and the connection with gradient descent methods is also discussed. Furthermore, a general algorithm for any data matrix is derived, and the computational complexity is analyzed.
4. Extensive experiments on real world datasets (including two document 85 datasets and three image datasets) demonstrate that the proposed GCCF algorithm shows strong robustness for clustering tasks, and achieves better performance when compared with several existing related methods.

The rest of the paper is organized as follows. In Section II, we briefly give a review of CF and LCCF, and then introduces the MCC and the half-quadratic technique. Section III develops the GCCF algorithm. The algorithm analysis

Table 1: Notations

Notations	Descriptions
M	the dimension of features
N	the number of samples
$K < \min(M, N)$	the embedding dimension of concept factorization
$\mathbf{L}$	the graph Laplacian matrix
λ	the regularization parameter
p	the number of nearest neighbors
$\ \cdot\ $	Frobenius norm

90 of GCCF is given in Section IV. Experimental results are then presented in Section V. Finally, Section VI draws the conclusion and some suggestions for future work.

For clarity, in this paper, capital boldface letters, small boldface letters, and normal font are respectively used to denote matrices, vectors, and scalars, e.g.,
95 $\mathbf{X}$, $\mathbf{x}$, and x . x_{ij} (or $\mathbf{X}_{ij}$) denotes the element in the i -th row and j -th column of matrix $\mathbf{X}$. In addition, some important notations are listed in Table I.

2. Related Work

2.1. CF

Suppose there is a data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}_{\geq 0}^{M \times N}$, where each column of $\mathbf{X}$ denotes a sample vector $\mathbf{x}_j$ containing M elements. The goal of NMF is to decompose $\mathbf{X}$ into two nonnegative matrices $\mathbf{U} = [u_{ik}] \in \mathbb{R}_{\geq 0}^{M \times K}$ and $\mathbf{V} = [v_{jk}] \in \mathbb{R}_{\geq 0}^{N \times K}$ such that the product of these two matrices can well approximate the original data matrix [12, 13]:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T \quad (1)$$

where $[\cdot]^T$ is the transpose operator. In fact, this approximation can also be expressed as

$$\mathbf{x}_j \approx \sum_{k=1}^K \mathbf{u}_k v_{jk} \quad (2)$$

where $\mathbf{u}_k$ denotes the k -th column vector of $\mathbf{U}$. Based on (2), one can observe that, each sample vector $\mathbf{x}_j$ can be approximated by a linear combination of the columns of $\mathbf{U}$, weighted by the components of j -th row vector of $\mathbf{V}$ [17, 18]. Therefore, $\mathbf{U}$ and $\mathbf{V}$ are called as the **basis matrix** and **coefficient matrix** respectively. Due to the property that NMF leads to a parts-based representation, NMF can achieve good performance in document and image clustering tasks. However, NMF still has some shortcomings. Specifically, NMF cannot be adopted in some applications when the data contains negative values. As an extension of NMF, CF was proposed by Xu and Gong [14], which inherits all the advantages of NMF and overcomes the drawback of NMF. Assume that $\mathbf{u}_k$ denotes the k -th column vector of $\mathbf{U}$. In the CF model, $\mathbf{u}_k$ can be represented by a linear combination of the N data points

$$\mathbf{u}_k = \sum_{j=1}^N \mathbf{x}_j w_{jk} \quad (3)$$

where $w_{jk} \geq 0$ and $\mathbf{W} = [w_{jk}] \in \mathbb{R}^{N \times K}$ is a **nonnegative association weight matrix** [14]. Therefore, CF tries to satisfy the following approximation:

$$\mathbf{X} \approx \mathbf{XWV}^T \quad (4)$$

The Frobenius norm $\|\mathbf{A}\|^2$ is commonly used to quantify the approximation, and accordingly the optimization problem of CF can be expressed as follows:

$$\min_{\mathbf{W}, \mathbf{V}} \mathcal{O}_{CF} = \frac{1}{2} \|\mathbf{X} - \mathbf{XWV}^T\|^2 \quad (5)$$

The multiplicative updating rules for minimizing the objective function $\mathcal{O}_{CF}$ are presented as follows:

$$w_{jk} \leftarrow w_{jk} \frac{(\mathbf{KV})_{jk}}{(\mathbf{KWV}^T \mathbf{V})_{jk}} \quad v_{jk} \leftarrow v_{jk} \frac{(\mathbf{KW})_{jk}}{(\mathbf{VW}^T \mathbf{KW})_{jk}} \quad (6)$$

where $\mathbf{K} = \mathbf{X}^T \mathbf{X}$.

The standard CF method shows the superior performance for solving many real world problems. However, CF only considers this learning in the Euclidean space, and ignores the intrinsic geometrical information of the raw data. Cai *et al.* exploited the intrinsic geometrical information of the data and incorporated it as an additional regularization term into the objective function, and proposed a locally consistent CF algorithm (LCCF). According to [21], the optimization problem of LCCF is given by

$$\min_{\mathbf{W}, \mathbf{V}} \mathcal{O}_{LCCF} = \frac{1}{2} \|\mathbf{X} - \mathbf{XWV}^T\|^2 + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \quad (7)$$

where $\text{Tr}(\cdot)$ is the trace of a matrix, $\mathbf{L} = \mathbf{D} - \mathbf{S}$, $\mathbf{S}$ denotes the edge weight matrix, and $\mathbf{D}$ stands for a diagonal matrix whose entries are column (or row) sums of $\mathbf{S}$. Cai *et al.* derived the following multiplicative updating rules:

$$w_{jk} \leftarrow w_{jk} \frac{(\mathbf{KV})_{jk}}{(\mathbf{KWV}^T \mathbf{V})_{jk}} \quad v_{jk} \leftarrow v_{jk} \frac{(\mathbf{KW} + \lambda \mathbf{SV})_{jk}}{(\mathbf{VW}^T \mathbf{KW} + \lambda \mathbf{DV})_{jk}} \quad (8)$$

In recent studies, a nearest neighbor graph has been frequently applied to model the local geometric structure of a scatter of data points [18, 21, 37, 38]. Assume that a graph contains N vertices in which each vertex corresponds to a sample of the data. For each data point $\mathbf{x}_j$, we obtain its p -nearest neighbors and put edges between $\mathbf{x}_j$ and its neighbors. Generally speaking, three common approaches are adopted to calculate the edge weight matrix $\mathbf{S}$ [17, 24]:

1. **0-1 weighting.**

$$\mathbf{S}_{js} = \begin{cases} 1, & \text{if } \mathbf{x}_s \in N_p(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases} \quad j, s = 1, \dots, N$$

where $N_p(\mathbf{x}_j)$ is the set of p -nearest neighbors of $\mathbf{x}_j$.

2. **Heat kernel weighting.**

$$\mathbf{S}_{js} = \exp^{-\frac{\|\mathbf{x}_j - \mathbf{x}_s\|^2}{2\sigma^2}} \quad j, s = 1, \dots, N$$

3. **Dot-product weighting.**

$$\mathbf{S}_{js} = \mathbf{x}_j^T \mathbf{x}_s \quad j, s = 1, \dots, N$$

It is worth noting that, when the sample data is normalized to 1, the dot product of two samples is equal to the cosine similarity of the two samples. Actually,
110 different similarity measures are chosen for different applications. For instance, the dot-product weighting may be a better choice for document clustering tasks, while the 0-1 weighting and the heat kernel weighting are frequently suggested for image clustering tasks.

2.3. Correntropy

In recent years, correntropy was proposed in ITL for handling non-Gaussian noises or outlier, which usually caused the unpredictable errors in many real world applications [25, 27, 28, 33, 34]. As a similarity measure between two random variables A and B , correntropy is directly related to Renyi's quadratic entropy in which Parzen windowing method is adopted to estimate the data's probability distribution, and defined by [26, 35, 39, 40]

$$V(A, B) = E[\kappa(A, B)] = \int \kappa(a, b) dF_{AB}(a, b) \quad (9)$$

where $E[\cdot]$ is the expectation operator, $\kappa(\cdot, \cdot)$ denotes a shift-invariant Mercer kernel, and $F_{AB}(a, b)$ stands for the joint distribution function of (A, B) . It takes the advantage of a kernel trick that nonlinearly maps the input space to a higher dimensional feature space. In this paper, without mentioning otherwise, the kernel function of correntropy $\kappa(\cdot, \cdot)$ is the Gaussian kernel, given by

$$\kappa_{\sigma}(a, b) = g(a - b) \triangleq \exp^{-\frac{(a-b)^2}{2\sigma^2}} \quad (10)$$

where $\sigma > 0$ denotes the kernel bandwidth parameter. In practice, the joint distribution $F_{AB}(a, b)$ is usually unknown, and only a finite number of data samples $\{(a_n, b_n)\}_{n=1}^N$ are available. In these situations, the correntropy can be estimated by

$$\hat{V}_{N,\sigma}(A, B) = \frac{1}{N} \sum_{n=1}^N g(a_n - b_n) \quad (11)$$

115 where $\hat{V}_{N,\sigma}$ stands for the estimator $\hat{V}$. The maximum of the correntropy of error in (11) is called maximum correntropy criterion (MCC). Fig. 1 shows

Figure 1: Cost functions in the joint space: (a) MSE; (b) MCC ($\sigma = 1.0$).

the cost functions of means square error (MSE) and MCC in the joint space. As one can see that, different from MSE, the MCC emphasizes the behavior along the line $a = b$ while reducing contributions away from it. Specifically, based on the geometric meaning, three regions (i.e., Euclidean region, transition
120 region and rectification region) are divided in Fig. 1(b). the MCC behaves like l_2 -norm distance in the Euclidean region, similarly like a l_1 -norm distance in the transition region and eventually approaches a l_0 -norm in the rectification region. Additionally, from the viewpoint of statistics, compared with the global
125 similarity measure MSE, which only quantifies the second-order moment, the MCC is a local metric, and can capture the higher order moments (e.g., all the even-order moments [26, 28, 35]). Therefore, due to the above reasons, MCC based methods have better performance in many practical applications, especially when the data involve non-Gaussian noises or outliers.

130 2.4. Half-Quadratic Minimization

Generally speaking, the objective function of ITL based methods is nonlinear and non-convex, and is difficult to optimize directly. Fortunately, the half-quadratic (HQ) technique has been frequently used to solve these ITL based optimization problems [25, 29, 33, 34]. Assume that $\phi(\mathbf{e})$ is a potential cost

function on a vector $\mathbf{e} \in \mathbb{R}^d$, defined as

$$\phi(\mathbf{e}) = \sum_{i=1}^d \phi(e_i) \quad (12)$$

where e_i denotes the i -th element in $\mathbf{e}$. Based on the HQ minimization, (12) is reformulated as an augmented objective function by introducing an auxiliary variable ϱ_i for a fixed e_i , which is given by [37]

$$\phi(e_i) = \min_{\varrho_i} \{Q(e_i, \varrho_i) + \varphi(\varrho_i)\} \quad (13)$$

where ϱ_i only depends on $\phi(\cdot)$ (see the specific form later), $Q(e_i, \varrho_i)$ stands for the HQ function, and $\varphi(\cdot)$ denotes the dual potential function of $\phi(\cdot)$. Previous studies show that two commonly used forms (i.e., the additive form and the multiplicative form) are adopted to formulate the HQ function. However, in the last decade, the multiplicative form has received considerable attention. Therefore, in this paper, we only consider the multiplicative form of $Q(e_i, \varrho_i)$, which can be expressed as follows:

$$Q(e_i, \varrho_i) = \frac{1}{2} \varrho_i e_i^2 \quad (14)$$

Assume that $Q(\mathbf{e}, \boldsymbol{\varrho}) = \sum_{i=1}^d Q(e_i, \varrho_i)$, where $\boldsymbol{\varrho} = [\varrho_1, \dots, \varrho_d]$. Combining (13) and (14), we obtain the vector form of (13):

$$\phi(\mathbf{e}) = \min_{\boldsymbol{\varrho}} \left\{ Q(\mathbf{e}, \boldsymbol{\varrho}) + \sum_{i=1}^d \varphi(\varrho_i) \right\} \quad (15)$$

3. Correntropy based Graph Regularized Concept Factorization

The performance of these CF based methods (e.g., the standard CF and LCCF) may degrade seriously, when the raw data involves different types of non-Gaussian noises or outliers. In this section, we propose the correntropy based graph regularized concept factorization (GCCF), which is insensitive to the non-Gaussian noises or outliers.

3.1. Objective Function

Based on MCC and the graph regularizer of data manifold, a robust correntropy based graph regularized concept factorization (GCCF) is proposed in this work, which maximizes the following objective function:

$$\max_{\mathbf{W}, \mathbf{V}} \mathcal{O}_{GCCF} = \sum_{i=1}^M g \left(\sqrt{\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{xwV}^T)_{ij} \right)^2} \right) - \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \quad (16)$$

Clearly, the objective function of GCCF in (16) is nonlinear and non-convex, and is hard to optimize directly. In order to solve the above optimization problem, the HQ technique is utilized to convert the non-convex term of the objective function into a quadratic term of the multiplicative form [27, 36]. Based on the property of the convex conjugate function, we have the proposition 1 [37, 41] in the following:

Proposition 1: *There exists a convex conjugate function of $g(z)$, such that*

$$g(z) = \max_{\varrho} \left(\varrho \frac{\|z\|^2}{\sigma^2} - \varphi(\varrho) \right) \quad (17)$$

and for a fix z , the maximum is reached at $\varrho = -g(z)$.

According to the proposition 1, the augmented objective function of GCCF in an enlarged parameter space is derived by substituting (17) into (16):

$$\begin{aligned} \max_{\mathbf{W}, \mathbf{V}, \boldsymbol{\varrho}} \hat{\mathcal{O}}_1 = & \sum_{i=1}^M \left(\frac{\varrho_i}{\sigma^2} \left(\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{xwV}^T)_{ij} \right)^2 \right) - \varphi(\varrho_i) \right) \\ & - \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \end{aligned} \quad (18)$$

where $\boldsymbol{\varrho} = [\varrho_1, \dots, \varrho_M]^T$ is the auxiliary vector in HQ optimization technique. Next we adopt the alternate minimization method to optimize the augmented objective function $\hat{\mathcal{O}}_1$. Specifically, for a fixed $\mathbf{W}$ and $\mathbf{V}$, maximizing the augmented objective function $\hat{\mathcal{O}}_1$ with respect to $\boldsymbol{\varrho}$, we have

$$\begin{aligned} \varrho_i = & -g \left(\sqrt{\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{xwV}^T)_{ij} \right)^2} \right) \\ = & -\exp \left(-\frac{\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{xwV}^T)_{ij} \right)^2}{2\sigma^2} \right) \end{aligned} \quad (19)$$

The kernel bandwidth parameter σ in (18) is a significant free parameter since it controls all robust properties of correntropy. An appropriate kernel bandwidth usually can provide an efficient mechanism to eliminate the influence of non-Gaussian noises and outliers. In many cases, the kernel bandwidth parameter is selected empirically. In this work, we calculate the parameter σ by the following equation [27]

$$\sigma = \sqrt{\frac{1}{2M} \sum_{i=1}^M \sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{x} \mathbf{W} \mathbf{V}^T)_{ij} \right)^2} \quad (20)$$

Then, for a fixed $\boldsymbol{\varrho}$, based on (18), we derive

$$\max_{\mathbf{W}, \mathbf{V}} \hat{\mathcal{O}}_1 = \sum_{i=1}^M \left(\frac{\varrho_i}{\sigma^2} \left(\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{x} \mathbf{W} \mathbf{V}^T)_{ij} \right)^2 \right) \right) - \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \quad (21)$$

which is also equivalent to the following optimization problem:

$$\min_{\mathbf{W}, \mathbf{V}} \hat{\mathcal{O}}_2 = \sum_{i=1}^M \left(h_i \left(\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{x} \mathbf{W} \mathbf{V}^T)_{ij} \right)^2 \right) \right) + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \quad (22)$$

145 where $h_i = -\frac{\varrho_i}{\sigma^2}$.

3.2. Optimization

Obviously, it is difficult to solve the optimization problem of GCCF directly, since the augmented objection function $\hat{\mathcal{O}}_2$ in (22) is non-convex with respect to $\mathbf{W}$ and $\mathbf{V}$ together. In this paper, we adopt an iterative method which can obtain a local minimum. Denote $\mathcal{R} = \mathbf{X}^T \mathbf{H} \mathbf{X}$, where $\mathbf{H}$ is a diagonal matrix whose diagonal element is $\mathbf{H}_{ii} = h_i$,

$$h_i = -\frac{\varrho_i}{\sigma^2} = \frac{\exp \left(-\frac{\sum_{j=1}^N \left(\mathbf{x}_{ij} - (\mathbf{x} \mathbf{W} \mathbf{V}^T)_{ij} \right)^2}{2\sigma^2} \right)}{\sigma^2} \quad (23)$$

We rewrite the $\hat{\mathcal{O}}_2$ in (22) as follows:

$$\begin{aligned} \hat{\mathcal{O}}_2 &= \text{Tr} \left(\left(\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T \right)^T \mathbf{H} \left(\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T \right) \right) + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &= \text{Tr} \left(\left(\mathbf{I} - \mathbf{W} \mathbf{V}^T \right)^T \mathcal{R} \left(\mathbf{I} - \mathbf{W} \mathbf{V}^T \right) \right) + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &\stackrel{(a)}{=} \text{Tr}(\mathcal{R}) - 2 \text{Tr}(\mathbf{V} \mathbf{W}^T \mathcal{R}) + \text{Tr}(\mathbf{V} \mathbf{W}^T \mathcal{R} \mathbf{W} \mathbf{V}^T) + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \end{aligned} \quad (24)$$

where (a) follows from the properties $Tr(\mathbf{AB}) = Tr(\mathbf{BA})$ and $Tr(\mathbf{A}^T) = Tr(\mathbf{A})$. Assume that Φ_{jk} and Ψ_{jk} are the Lagrange multiplier with constraints $\Phi_{jk} \geq 0$ and $\Psi_{jk} \geq 0$ respectively. Then the Lagrange function $\mathcal{L}$ is obtained by defining the matrices $\Phi = [\Phi_{jk}] \in \mathbb{R}_{\geq 0}^{N \times K}$ $\Psi = [\Psi_{jk}] \in \mathbb{R}_{\geq 0}^{N \times K}$

$$\begin{aligned} \mathcal{L} = & Tr(\mathcal{R}) - 2Tr(\mathbf{V}\mathbf{W}^T\mathcal{R}) + Tr(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W}\mathbf{V}^T) + \\ & \lambda Tr(\mathbf{V}^T\mathbf{L}\mathbf{V}) + Tr(\Phi\mathbf{W}^T) + Tr(\Psi\mathbf{V}^T) \end{aligned} \quad (25)$$

Then, similar to [21], using the partial derivative of $\mathcal{L}$ with respect to $\mathbf{W}$ and $\mathbf{V}$, we derive

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = -2\mathcal{R}\mathbf{V} + 2\mathcal{R}\mathbf{W}\mathbf{V}^T\mathbf{V} + \Phi \quad (26)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = -2\mathcal{R}\mathbf{W} + 2\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} + 2\lambda\mathbf{L}\mathbf{V} + \Psi \quad (27)$$

Using the Karush-Kuhn-Tucker conditions (i.e., $\Phi_{jk}w_{jk} = 0$ and $\Psi_{jk}v_{jk} = 0$) [42, 43], we have

$$-2(\mathcal{R}\mathbf{V})_{jk}w_{jk} + 2(\mathcal{R}\mathbf{W}\mathbf{V}^T\mathbf{V})_{jk}w_{jk} + \Phi_{jk}w_{jk} = 0 \quad (28)$$

$$-2(\mathcal{R}\mathbf{W})_{jk}v_{jk} + 2(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W})_{jk}v_{jk} + 2(\lambda\mathbf{L}\mathbf{V})_{jk}v_{jk} + \Psi_{jk}v_{jk} = 0 \quad (29)$$

which lead to the following equations:

$$w_{jk} \leftarrow w_{jk} \frac{(\mathcal{R}\mathbf{V})_{jk}}{(\mathcal{R}\mathbf{W}\mathbf{V}^T\mathbf{V})_{jk}} \quad (30)$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathcal{R}\mathbf{W} + \lambda\mathbf{S}\mathbf{V})_{jk}}{(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} + \lambda\mathbf{D}\mathbf{V})_{jk}} \quad (31)$$

Noted that, if $\mathbf{W}$ and $\mathbf{V}$ are the solution for the proposed GCCF algorithm, then $\mathbf{W}\mathbf{Y}^{-1}$ and $\mathbf{V}\mathbf{Y}$ are also a solution for any diagonal matrix $\mathbf{Y}$ with positive diagonal elements due to $(\mathbf{W}\mathbf{Y}^{-1})(\mathbf{V}\mathbf{Y})^T = \mathbf{W}\mathbf{V}^T$. To overcome this problem, in practical applications, one strategy is usually adopted to normalize each row of $\mathbf{W}$ to be 1 [44, 45]:

$$w_{jk} \leftarrow \frac{w_{jk}}{\sqrt{\sum_{k=1}^K w_{jk}^2}}, \quad v_{jk} \leftarrow v_{jk} \sqrt{\sum_{k=1}^K w_{jk}^2} \quad (32)$$

Algorithm 1 GCCF Algorithm

Input: The data matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{M \times N}$, the initial matrices $\mathbf{W} \in \mathbb{R}_{\geq 0}^{N \times K}$, $\mathbf{V} \in \mathbb{R}_{\geq 0}^{N \times K}$, the regularization parameter λ .

Output: two factor matrices $\mathbf{W}$, $\mathbf{V}$.

- 1: **repeat**
 - 2: Update σ by using (20);
 - 3: Update $\mathbf{H}$ by using (23);
 - 4: Update $\mathbf{W}$ by using (30);
 - 5: Update $\mathbf{V}$ by using (31);
 - 6: **until** Converges
-

Furthermore, based on the multiplicative updates, the detailed steps for GCCF are shown in Algorithm 1.

150 4. Algorithm Analysis

4.1. Convergence Analysis

Similar to the convergence analysis of LCCF algorithm in [21], we prove that the objective function $\hat{\mathcal{O}}_2$ in (24) is non-increasing under the update rules in (30) and (31), by using an auxiliary function that was used in the Expectation-
155 Maximization algorithm [46]. Firstly, we introduce the definition of the auxiliary function, and the lemma 1 [16].

Definition $G(v, v')$ is an auxiliary function for $F(v)$ if the following conditions

$$G(v, v') \geq F(v), \quad G(v, v) = F(v) \quad (33)$$

are satisfied.

Lemma 1 If G is an auxiliary function for F , then F is non-increasing under the following update steps from iteration t to $t + 1$:

$$v^{t+1} = \arg \min_v G(v, v^t) \quad (34)$$

Proof:

$$F(v^{t+1}) \leq G(v^{t+1}, v^t) \leq G(v^t, v^t) = F(v^t)$$

Next, a proper auxiliary function for the objective function $\hat{\mathcal{O}}_2$ will be defined. By using the Lemma 1, we show that the update rule for minimizing the auxiliary function G is exactly the update rule (31). In this situation, we conclude that the objective function $\hat{\mathcal{O}}_2$ is non-increasing under the update rule (31). Assume that v_{jk} denotes any element in $\mathbf{V}$, and F_{jk} is the part of $\hat{\mathcal{O}}_2$ which only related to v_{jk} . Then we have

$$F'_{jk} = \left(\frac{\partial \hat{\mathcal{O}}_2}{\partial \mathbf{V}} \right)_{jk} = \left(-2\mathcal{R}\mathbf{W} + 2\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} + 2\lambda\mathbf{L}\mathbf{V} \right)_{jk} \quad (35)$$

$$F''_{jk} = \left(\frac{\partial^2 \hat{\mathcal{O}}_2}{\partial \mathbf{V}^2} \right)_{jk} = \left(2\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{kk} + (2\lambda\mathbf{L})_{jj} \quad (36)$$

Since our update rule is element-wise, we need to prove that each F_{jk} is non-increasing under (31).

Lemma 2 The following function

$$G(v, v_{jk}^t) = F_{jk}(v_{jk}^t) + F'_{jk}(v_{jk}^t)(v - v_{jk}^t) + \frac{\left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{jk} + (\lambda\mathbf{D}\mathbf{V})_{jk}}{v_{jk}^t} (v - v_{jk}^t)^2 \quad (37)$$

is an auxiliary function for F_{jk} , which is only relevant to v_{jk} .

Proof: Obviously, $G(v, v) = F_{jk}(v)$. Next we prove $G(v, v_{jk}^t) \geq F_{jk}(v)$. By expanding the Taylor series of $F_{jk}(v)$ and using (36), we derive

$$\begin{aligned} F_{jk}(v) &= F_{jk}(v_{jk}^t) + F'_{jk}(v_{jk}^t)(v - v_{jk}^t) + \frac{1}{2}F''_{jk}(v_{jk}^t)(v - v_{jk}^t)^2 \\ &= F_{jk}(v_{jk}^t) + F'_{jk}(v_{jk}^t)(v - v_{jk}^t) + \left(\left(\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{kk} + (\lambda\mathbf{L})_{jj} \right) (v - v_{jk}^t)^2 \end{aligned} \quad (38)$$

Combining (37) and (38), to ensure that $G(v, v_{jk}^t) \geq F_{jk}(v)$ is equivalent to

$$\frac{\left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{jk} + (\lambda\mathbf{D}\mathbf{V})_{jk}}{v_{jk}^t} \geq \left(\left(\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{kk} + (\lambda\mathbf{L})_{jj} \right) \quad (39)$$

Since

$$\left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W}\right)_{jk} = \sum_{l=1}^K v_{jl}^t \left(\mathbf{W}^T\mathcal{R}\mathbf{W}\right)_{lk} \geq v_{jk}^t \left(\mathbf{W}^T\mathcal{R}\mathbf{W}\right)_{kk} \quad (40)$$

and

$$\begin{aligned} (\lambda\mathbf{D}\mathbf{V})_{jk} &= \lambda \sum_{i=1}^N \mathbf{D}_{ji} v_{ik} \geq \lambda \mathbf{D}_{jj} v_{jk}^t \\ &\geq \lambda (\mathbf{D} - \mathbf{S})_{jj} v_{jk}^t = \lambda \mathbf{L}_{jj} v_{jk}^t \end{aligned} \quad (41)$$

hence (39) holds and $G(v, v_{jk}^t) \geq F_{jk}(v)$. Replacing $G(v, v_{jk}^t)$ in (34) by (37), we have

$$\begin{aligned} v_{jk}^{t+1} &= v_{jk}^t - v_{jk}^t \frac{F'_{jk}(v_{jk}^t)}{2 \left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W}\right)_{jk} + 2(\lambda\mathbf{D}\mathbf{V})_{jk}} \\ &= v_{jk}^t \frac{(\mathcal{R}\mathbf{W} + \lambda\mathbf{S})_{jk}}{\left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} + \lambda\mathbf{D}\mathbf{V}\right)_{jk}} \end{aligned} \quad (42)$$

In a similar way, one can also prove that the objective function $\hat{\mathcal{O}}_2$ is non-increasing under (30).

4.2. Connection with Gradient Descent Method

Instead of using the multiplicative update approach to obtain the matrix factors, gradient descent method has also been successfully applied to solve the optimization problems [47]. Based on gradient descent method, we can derive the following additive update rules for the objective function $\hat{\mathcal{O}}_2$:

$$\begin{aligned} w_{jk} &= w_{jk} + \xi_{jk} \frac{\partial \hat{\mathcal{O}}_2}{\partial w_{jk}} \\ &= w_{jk} + \xi_{jk} \left(-2(\mathcal{R}\mathbf{V})_{jk} + 2 \left(\mathcal{R}\mathbf{W}\mathbf{V}^T\mathbf{V} \right)_{jk} \right) \end{aligned} \quad (43)$$

$$\begin{aligned} v_{jk} &= v_{jk} + \eta_{jk} \frac{\partial \hat{\mathcal{O}}_2}{\partial v_{jk}} = v_{jk} + \\ &\eta_{jk} \left(-2(\mathcal{R}\mathbf{W})_{jk} + 2 \left(\mathbf{V}\mathbf{W}^T\mathcal{R}\mathbf{W} \right)_{jk} + 2\lambda(\mathbf{L}\mathbf{V})_{jk} \right) \end{aligned} \quad (44)$$

where ξ_{jk} and η_{jk} are the step size parameters of gradient descent method. Generally speaking, if ξ_{jk} and η_{jk} are enough small, the above updates should reduce $\hat{\mathcal{O}}_2$. When $\xi_{jk} = -w_{jk}/2 \left(\mathcal{R} \mathbf{W} \mathbf{V}^T \mathbf{V} \right)_{jk}$ and $\eta_{jk} = -v_{jk}/2 \left(\mathbf{V} \mathbf{W}^T \mathcal{R} \mathbf{W} + \lambda \mathbf{D} \mathbf{V} \right)_{jk}$, using $\mathbf{L} = \mathbf{D} - \mathbf{S}$, we derive

$$\begin{aligned} w_{jk} &= w_{jk} - \frac{w_{jk}}{2 \left(\mathcal{R} \mathbf{W} \mathbf{V}^T \mathbf{V} \right)_{jk}} \left(-2 \left(\mathcal{R} \mathbf{V} \right)_{jk} + 2 \left(\mathcal{R} \mathbf{W} \mathbf{V}^T \mathbf{V} \right)_{jk} \right) \\ &= w_{jk} \frac{(\mathcal{R} \mathbf{V})_{jk}}{(\mathcal{R} \mathbf{W} \mathbf{V}^T \mathbf{V})_{jk}} \end{aligned} \quad (45)$$

$$\begin{aligned} v_{jk} &= v_{jk} - \frac{v_{jk}}{2 \left(\mathbf{V} \mathbf{W}^T \mathcal{R} \mathbf{W} + \lambda \mathbf{D} \mathbf{V} \right)_{jk}} \times \\ &\quad \left(-2 \left(\mathcal{R} \mathbf{W} \right)_{jk} + 2 \left(\mathbf{V} \mathbf{W}^T \mathcal{R} \mathbf{W} \right)_{jk} + 2\lambda \left(\mathbf{L} \mathbf{V} \right)_{jk} \right) \\ &= v_{jk} \frac{(\mathcal{R} \mathbf{W} + \lambda \mathbf{S} \mathbf{V})_{jk}}{(\mathbf{V} \mathbf{W}^T \mathcal{R} \mathbf{W} + \lambda \mathbf{D} \mathbf{V})_{jk}} \end{aligned} \quad (46)$$

Obviously, using some tricks to select the step size parameters in gradient descent method, the multiplicative updating rules (i.e., (30) and (31)) can be considered as the special case of gradient descent method. Indeed, the main reason for adopting the multiplicative updating rules is to ensure the nonnegativity of the matrix factors.

4.3. Algorithm for Negative Data Matrices

The proposed GCCF method only employs for the case the data matrix is nonnegative. However, the data matrix may contain the negative entries in many practical applications. In this subsection, we derive a general algorithm for any case. Similar to [14, 21], the algorithm is proposed based on the following theorem [48].

Theorem 1: Define the nonnegative general quadratic form as

$$f(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T \mathbf{A} \mathbf{v} + \mathbf{b}^T \mathbf{v} \quad (47)$$

where $\mathbf{v} = [v_i]$ denotes an $m \times 1$ nonnegative vector, $\mathbf{A}$ stands for an arbitrary $m \times m$ symmetric semipositive definite matrix, and $\mathbf{b} = [b_i]$ is a $m \times 1$ vector. Assume that $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$, where

$$\mathbf{A}_{ij}^+ = \begin{cases} \mathbf{A}_{ij}, & \text{if } \mathbf{A}_{ij} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \mathbf{A}_{ij}^- = \begin{cases} |\mathbf{A}_{ij}|, & \text{if } \mathbf{A}_{ij} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

Then the solution $\mathbf{v}$ for minimizing $f(\mathbf{v})$ can be obtained:

$$v_i \leftarrow v_i \left[\frac{-b_i + \sqrt{b_i^2 + 4(\mathbf{A}^+ \mathbf{v})_i (\mathbf{A}^- \mathbf{v})_i}}{2(\mathbf{A}^+ \mathbf{v})_i} \right] \quad (48)$$

It is clearly seen that, fixing $\mathbf{V}$, the objective function $\hat{\mathcal{O}}_2$ is a quadratic form of $\mathbf{W}$. Therefore, we can apply the Theorem 1 to the $\hat{\mathcal{O}}_2$, and calculate the corresponding coefficients $\mathbf{b}$ and $\mathbf{A}$. Based on the first-order and second-order derivatives with respect to $\mathbf{W} = 0$ and $\mathbf{W}$, we have

$$\left. \frac{\partial \hat{\mathcal{O}}_2}{\partial w_{jk}} \right|_{\mathbf{W}=0} = -2(\mathcal{R}\mathbf{V})_{jk} \quad (49)$$

$$\frac{\partial^2 \hat{\mathcal{O}}_2}{\partial w_{jk} \partial w_{il}} = 2\mathcal{R}_{ji}(\mathbf{V}^T \mathbf{V})_{lk} = 2\mathcal{R}_{ji}^+(\mathbf{V}^T \mathbf{V})_{lk} - 2\mathcal{R}_{ji}^-(\mathbf{V}^T \mathbf{V})_{lk} \quad (50)$$

where $\mathcal{R} = \mathcal{R}^+ - \mathcal{R}^-$, and

$$\mathcal{R}_{ij}^+ = \begin{cases} \mathcal{R}_{ij}, & \text{if } \mathcal{R}_{ij} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \mathcal{R}_{ij}^- = \begin{cases} |\mathcal{R}_{ij}|, & \text{if } \mathcal{R}_{ij} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

Substituting (49) and (50) into (48) respectively, the multiplicative updating rule for $\mathbf{W}$ is obtained

$$w_{jk} \leftarrow w_{jk} \frac{(\mathcal{R}\mathbf{V})_{jk} + \sqrt{(\mathcal{R}\mathbf{V})_{jk}^2 + 4\mathbf{P}_{jk}^+ \mathbf{P}_{jk}^-}}{2\mathbf{P}_{jk}^+} \quad (51)$$

where $\mathbf{P}^+ = \mathcal{R}^+ \mathbf{W} \mathbf{V}^T \mathbf{V}$ and $\mathbf{P}^- = \mathcal{R}^- \mathbf{W} \mathbf{V}^T \mathbf{V}$. Similarly, fixing $\mathbf{W}$, we have

$$\left. \frac{\partial \hat{\mathcal{O}}_2}{\partial v_{jk}} \right|_{\mathbf{V}=0} = -2(\mathcal{R}\mathbf{W})_{jk} \quad (52)$$

$$\frac{\partial^2 \hat{\mathcal{O}}_2}{\partial v_{jk} \partial v_{il}} = 2\delta_{ij}(\mathbf{W}^T \mathcal{R} \mathbf{W})_{lk} + 2\lambda \delta_{lk} \mathbf{L}_{ji} \quad (53)$$

where δ_{ij} (or δ_{lk}) is equal to 1 if $i = j$ (or $l = k$) and equal to 0 otherwise. Substituting (52) and (53) into (48), we have the multiplicative updating rule for $\mathbf{V}$:

$$v_{jk} \leftarrow v_{jk} \frac{(\mathcal{R} \mathbf{W})_{jk} + \sqrt{(\mathcal{R} \mathbf{W})_{jk}^2 + 4\mathbf{Q}_{jk}^+ \mathbf{Q}_{jk}^-}}{2\mathbf{Q}_{jk}^+} \quad (54)$$

where $\mathbf{Q}^+ = \mathbf{V} \mathbf{W}^T \mathcal{R}^+ \mathbf{W} + \lambda \mathbf{D} \mathbf{V}$ and $\mathbf{Q}^- = \mathbf{V} \mathbf{W}^T \mathcal{R}^- \mathbf{W} + \lambda \mathbf{S} \mathbf{V}$.

180 4.4. Computational Complexity Analysis

The computational complexity is frequently used to evaluate the quality of an algorithm. In this subsection, we provide the computational complexity analysis for the proposed GCCF method. Based on the multiplicative updating rules in (30) and (31), it is obviously observed that the computational complexity is $O(N^2 K)$. In order to construct the p -nearest neighbor graph, the arithmetic operation in GCCF is $O(N^2 M + N^2 p)$. GCCF requires $O(MNK)$ to calculate the kernel bandwidth σ and $\mathbf{H}$ in (20) and (23) respectively. In addition, the proposed method needs to computer $\mathcal{R}$, which requires $O(N^2 M)$. Assume that T is the iteration number and $M \gg p$. Then the computational cost of GCCF is

$$O(T(N^2 K + MNK + N^2 M)) \quad (55)$$

Clearly, the computational complexity of GCCF is higher than the computational complexities of LCCF [21] and GCF [22]. Therefore, how to reduce the computational complexity of GCCF is an interesting topic for future study.

185 5. Experimental Results

In this section, some experiments are presented to show the effectiveness of the proposed GCCF method on document and image clustering applications in comparison to several related methods.

5.1. Datasets

190 Five datasets¹, such as Topic Detection Tracking (TDT2), Reuters, COIL20,
AR and CMU PIE, are used in the experiments. Among them, the first two
are the document datasets, the third is the object image dataset, and the last
two are the human face image datasets. Some samples from image datasets are
shown in Fig. 2. The brief descriptions of these datasets are given as follows
195 (also see Table 2):

1. The TDT2 dataset contains news data collected daily from six news agencies including CNN, NYT, VOA, ABC, APW, and PRI, over a period of the first half of 1998. The TDT2 dataset has 11201 on-topic documents which can be classified into 96 semantic categories. In our
200 experiment, we remove those categories with less than 30 documents and exclude those documents appearing in two or more categories. In such situation, a dataset that consists of 30 categories with a total of 9394 documents is left.
2. The Reuters dataset consists of 135 documents clusters with a total of
205 21578 documents. Generally speaking, the Reuters dataset is more difficult for clustering in comparison to the TDT2 dataset, since many documents in the Reuters dataset have multiple category labels and a broader variety of content. In our experiment, the documents with a unique category labels are kept, and the clusters with less than 10 are
210 discarded. This leaves us with 8213 documents in total.
3. The COIL20 dataset contains 1440 gray scale images of 72 objects viewed from different angles. Each object has 72 facial images, and each image is resized to 32×32 pixels.
4. The AR dataset² [49] includes over 4000 face images from 126 subjects
215 (56 women and 70 men). Each subject contains 26 facial images which are taken in two separate sessions. These images suffer from different

¹<http://www.cad.zju.edu.cn/home/dengcai/Data/data.html>

²<http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>

Table 2: Statistics of the datasets

Datasets	Domain	Instances	Features	Classes
TDT2	Document	9394	36771	30
Reuters	Document	8213	18933	41
COIL20	Image	1440	1024	20
AR	Image	1300	1024	100
CMU PIE	Image	2856	1024	68

facial variations, such as illumination variations (left light on, right light on, and all side lights on), occlusions by sunglasses or scarf, and various facial expressions (neural, smile, anger, and scream). Each image is resized to 32×32 pixels. In the experiments, we choose a subset of the dataset consisting of 50 male subjects and 50 female subjects in the first session.

5. The CMU PIE dataset contains 2856 gray scale images from 68 individuals. Each person has 42 facial images under different illumination and light conditions, and each image is resized to 32×32 pixels.

5.2. Evaluation Metric

To evaluate the clustering performance of all methods, three standard measures, such as Accuracy, Normalized Mutual Information (NMI), and Purity, are used in our experiments.

Accuracy is defined as follows:

$$\text{Accuracy} = \frac{\sum_{j=1}^N \delta(r_j, \text{map}(z_j))}{N} \quad (56)$$

where N denotes the total number of a dataset, r_j and z_j are respectively the label provided by the dataset and the cluster label, $\delta(r_j, z_j)$ stands for the delta function that equals 1 if $r_j = z_j$ and equals 0 otherwise, and $\text{map}(\cdot)$ is the mapping function that map each cluster label to the equivalent label from the data set by using the Kuhn-Munkres algorithm [50].

(a)

(b)

(c)

Figure 2: Sample images from three image datasets: (a) COIL20; (b) AR; (c) CMU PIE.

NMI is defined as follows:

$$\text{NMI} = \frac{\text{MI}(C, C')}{\max(H(C), H(C'))} \quad (57)$$

where C denotes the set of clusters from the ground truth, C' is the set of clusters provided by a clustering method, $H(C)$ and $H(C')$ stand for the entropies of the clusters C and C' , and $\text{MI}(C, C')$ denotes the mutual information (MI) of C and C' . Usually, $\text{MI}(C, C')$ is defined as

$$\text{MI} = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)} \quad (58)$$

where $p(c_i)$, $p(c'_j)$ and $p(c_i, c'_j)$ denote the probabilities that a data point arbitrarily selected from the dataset belongs to the clusters C , C' and both, respectively.

Purity is defined as follows:

$$\text{Purity} = \frac{1}{N} \sum_{j=1}^N \max(n_i^j) \quad (59)$$

where n_i^j denotes the number of samples in cluster i that belongs to original class j . Therefore, the purity is a real number in the interval $[0,1]$. The larger
240 the purity, the better the clustering performance.

5.3. Compared Algorithms and Experiment Settings

We compare the proposed method with seven related methods, which are listed below:

1. The standard k -means method (Kmeans).
- 245 2. The original nonnegative matrix factorization method (NMF)[12].
3. The original concept factorization method (CF)[14].
4. The locally consistent concept factorization method (LCCF)[21].
5. The dual-graph regularized concept factorization method (GCF)[22].
6. The adaptive dual-graph regularized CF with feature selection (ADGCF)
250 [51].
7. The graph regularized multilayer concept factorization method (GMCF)[52].

Especially, the clustering process is based on Kmeans clustering for all methods. In these experiments, there are two important parameters that we
255 need to set in advance for all graph based methods. Specifically, the number of nearest neighbors p and the regularization parameter λ are empirically set to be $p = 5$ and $\lambda = 100$ respectively [17, 21, 22]. The kernel bandwidth parameter σ is calculated by (20). Up to now, there is no systematic method for choosing the dimension parameter K . However, as discussed in [14, 17], in
260 order to accurately cluster the given dataset which consists of K clusters, it is better to project the data matrix into a K -dimensional subspace. In this situation, each axis corresponds to a particular cluster. Therefore, in this

work, we set the parameter K to be the number of clusters. Additionally, the evaluations are conducted under various cluster numbers. For each given cluster number, we randomly select the required samples from the original dataset, and experiment results are obtained by computing the average over 20 independent Monte Carlo runs with different initial values. For all datasets, the best result is bold-faced for each cluster number. As mentioned in Section II-B, for document and image clustering tasks, we respectively select the dot-product weighting and the 0-1 weighting approach to calculate the edge weight matrix.

5.4. Results Analysis

5.4.1. Clustering Results

Tables 3, 4, 5, 6, and 7 demonstrate the Accuracy, NMI, and Purity for different methods on the TDT2, Reuters, COIL20, AR, and CMU PIE datasets respectively. From these clustering results, one can observe that: i) the proposed GCCF method achieves better performance than all the other methods in comparison on all of these datasets in average. For example, in TDT2 dataset, compared with ADGCF which achieves the second best performance, GCCF can achieve 10.92%, 7.19%, and 11.94% improvement on the metrics accuracy, NMI, and purity respectively, in average. The main reason is that, the HQ optimization technique is adopted to convert the original correntropy based objective function into an iteratively reweighted optimization problem. In this case, the optimization of the GCCF method provides small feature weights to these noise samples and emphasizes on these noise-free samples. Therefore, the GCCF method outperforms other methods for clustering tasks. Furthermore, GCCF outperforms other methods on the AR dataset with sunglasses occlusion and scarf occlusion, which indicates that the GCCF method is robust to apparent outliers; ii) LCCF, GCF, ADGCF, GMCF, and GCCF consider the geometrical structure information of the data. These methods have better performance in comparison to Kmeans, CF, and NMF on the TDT2, Reuters and COIL20 datasets. This shows the importance

of the manifold structure of the data for clustering applications; iii) when the data (i.e., the AR dataset) contains the apparent outliers, GCCF still performs the best while LCCF and GCF work poorly since they are sensitive to the non-Gaussian noises. This case illustrates that the graph regularization item has little effect on outliers.

5.4.2. Parameters Selection

Two important parameters, the number of nearest neighbors p and the regularization parameter λ , are employed in the GCCF method. Indeed, up to now, there is no feasible strategy to calculate the optimal parameters (e.g., p and λ). Therefore, it is necessary to discuss how sensitive the proposed algorithm is to these parameters. Fig. 3 and Fig. 4 show the clustering results with different p and λ for five datasets respectively. In this experiment, when we vary the value of the parameter p from $\{3, 5, 7, 9, 11\}$, we fix the parameter λ at 100. Also when we vary the value of the parameter λ from $\{1, 10, 100, 1000, 10000\}$, we fix the parameter p at 5. As one can see that, if the parameter p (or λ) is too large or too small, the clustering performance of GCCF may become poor on the Reuters, COIL20, AR, and CMU PIE datasets. The proposed method achieves good performance on the TDT2 dataset when the parameter p (or λ) is enough large. The GCCF method has good performance with the parameter p varying from 5 to 9 and the parameter λ varying from 10 to 1000.

5.4.3. Convergence Study

Essentially, the updating steps of GCCF are derived by minimizing the objective function $\hat{\mathcal{O}}_2$, and hence, are iteratively approximating the local optimum. In Section IV-A, we have proved that GCCF is non-increasing. Here, we test the convergence of the proposed algorithm. Fig. 5 demonstrates the convergence curves of GCCF on all datasets. In each figure, the x-axis stands for the iteration number and the y-axis denotes the value of the objective function $\hat{\mathcal{O}}_2$. From experiment results, we observe that the GCCF

method is convergent before 200 iterations.

5.4.4. *Time Comparison*

In this subsection, we illustrate the average computational time for different
325 methods on five datasets. In this experiment, all methods are conducted on
an Inter Xeon E5-2697 V4 2.30GHz Windows Server with 128G memory. The
number of clusters is 5 for all datasets. We only calculate the time cost of the
clustering step for K-means, while for other methods, we calculate the time cost
of the decomposition and clustering step. Experimental results are listed in
330 Table 8. As one can see, GCCF needs the second longest time on all datasets.
The reason is that, at each iteration, GCCF requires to calculate the kernel
bandwidth σ and the diagonal matrix $\mathbf{H}$.

Table 3: Clustering result on TDT2

Methods	Accuracy (%)						
	5	10	15	20	25	30	Avg.
Kmeans	72.68	68.30	66.28	64.11	64.50	62.65	66.42
NMF	75.26	62.75	61.33	57.98	53.15	54.23	60.78
CF	77.10	63.49	66.25	62.18	53.12	56.90	63.17
LCCF	86.15	73.05	77.30	72.50	65.72	66.22	73.49
GCF	89.47	73.26	73.53	72.30	68.42	66.09	73.84
ADGCF	91.92	84.15	80.93	76.20	74.93	73.32	80.24
GMCF	90.52	82.14	80.09	77.17	73.72	71.72	79.22
GCCF	98.63	94.07	93.91	89.14	86.84	84.39	91.16

Methods	Normalized Mutual Information (%)						
	5	10	15	20	25	30	Avg.
Kmeans	62.69	69.73	74.71	74.24	76.74	75.31	72.23
NMF	65.33	64.39	69.57	68.25	67.16	67.29	66.99
CF	63.92	66.15	72.91	70.58	65.94	67.94	67.90
LCCF	75.16	72.17	80.41	77.42	75.39	75.77	76.05
GCF	84.27	74.46	80.15	77.85	76.99	76.14	78.31
ADGCF	87.54	80.07	81.09	79.96	78.29	78.33	80.88
GMCF	85.38	78.46	79.89	78.57	77.19	76.89	79.39
GCCF	95.55	89.17	91.15	85.56	84.40	82.61	88.07

Methods	Purity (%)						
	5	10	15	20	25	30	Avg.
Kmeans	75.39	73.50	71.05	69.45	69.33	67.70	71.07
NMF	79.47	69.71	68.13	64.71	61.92	62.79	67.78
CF	82.25	69.84	75.10	69.81	61.56	64.56	70.52
LCCF	91.30	78.42	81.77	76.22	71.41	72.23	78.55
GCF	90.40	76.43	76.68	74.96	73.13	70.85	77.07
ADGCF	92.22	86.56	82.89	80.71	80.49	76.95	83.30
GMCF	91.52	83.33	80.38	79.27	78.97	76.72	81.69
GCCF	99.61	95.85	97.95	94.36	92.33	91.37	95.24

Table 4: Clustering result on Reuters

Methods	Accuracy (%)								
	5	10	15	20	25	30	35	40	Avg.
Kmeans	55.75	46.75	35.70	35.35	31.50	29.16	31.50	26.50	36.52
NMF	57.98	45.52	27.94	26.16	30.33	28.95	29.40	27.19	34.18
CF	58.18	39.68	29.66	35.04	30.56	33.14	33.75	32.79	36.60
LCCF	66.26	59.44	43.29	55.75	53.75	37.18	38.37	33.47	48.43
GCF	67.41	59.47	38.73	42.10	41.54	30.64	30.14	27.61	42.21
ADGCF	75.92	64.29	56.27	50.22	42.39	40.61	41.44	37.80	51.11
GMCF	76.36	65.45	58.41	55.03	49.34	41.64	42.89	41.40	53.81
GCCF	78.96	68.86	68.01	68.40	60.45	59.61	52.60	50.51	63.42

Methods	Normalized Mutual Information (%)								
	5	10	15	20	25	30	35	40	Avg.
Kmeans	41.44	42.82	37.63	37.89	39.95	39.23	42.15	39.64	40.09
NMF	38.06	39.81	28.80	32.18	38.34	34.35	37.85	36.37	35.72
CF	37.08	36.99	28.68	33.15	36.97	32.74	35.87	34.86	34.54
LCCF	46.56	50.52	39.21	48.49	50.51	41.71	45.20	41.81	45.50
GCF	47.60	51.48	34.82	42.40	43.28	38.31	39.89	37.84	41.95
ADGCF	54.66	54.33	29.08	48.48	43.73	37.64	41.25	40.73	43.73
GMCF	52.41	53.28	29.22	34.21	50.50	48.09	41.91	41.56	43.89
GCCF	62.13	54.43	50.27	53.22	43.06	46.16	45.45	43.22	49.74

Methods	Purity (%)								
	5	10	15	20	25	30	35	40	Avg.
Kmeans	59.58	52.27	41.01	41.29	36.92	32.88	35.64	31.03	41.32
NMF	66.78	53.65	37.80	34.27	36.56	36.02	37.28	34.65	42.12
CF	69.25	50.04	38.99	44.29	38.36	41.25	44.28	42.63	46.13
LCCF	72.97	69.50	49.61	62.82	60.41	42.26	44.54	38.95	55.13
GCF	73.99	70.44	44.62	49.14	46.08	34.32	34.97	30.99	48.06
ADGCF	80.04	75.32	58.28	56.46	45.97	43.15	44.68	41.81	55.71
GMCF	81.30	75.12	59.90	58.14	53.98	45.81	45.91	44.39	58.06
GCCF	86.37	78.72	76.52	78.42	71.23	65.61	59.43	55.81	71.51

Table 5: Clustering result on COIL20

Methods	Accuracy (%)									
	4	6	8	10	12	14	16	18	20	Avg.
Kmeans	70.18	77.10	74.41	67.38	67.90	53.11	67.01	55.67	55.26	65.33
NMF	80.60	75.88	72.61	63.19	65.98	58.37	65.95	54.03	55.32	65.77
CF	78.91	72.31	70.19	72.03	59.59	58.18	61.14	56.39	51.60	64.48
LCCF	88.76	76.29	81.35	72.05	68.72	65.38	69.64	64.24	63.27	72.18
GCF	96.71	89.16	87.96	70.74	68.18	69.41	70.31	58.07	53.62	73.79
ADGCF	97.22	91.49	87.58	85.55	83.33	80.25	82.63	79.93	79.37	85.26
GMCF	96.52	90.97	89.93	84.44	79.39	80.85	83.16	75.54	75.13	83.99
GCCF	98.70	96.22	92.70	88.68	89.58	80.65	83.53	81.79	80.13	87.99

Methods	Normalized Mutual Information (%)									
	4	6	8	10	12	14	16	18	20	Avg.
Kmeans	58.82	71.06	83.42	74.69	76.63	67.78	76.46	71.45	70.13	72.27
NMF	71.62	71.38	79.06	66.79	73.47	69.48	74.87	70.58	70.63	71.98
CF	71.77	69.97	71.61	72.26	69.33	67.63	71.24	66.86	63.57	69.36
LCCF	78.34	79.88	84.37	84.35	79.62	74.81	80.39	76.54	76.95	79.47
GCF	95.66	84.05	81.27	80.88	76.04	73.84	82.80	70.19	67.24	79.10
ADGCF	97.22	92.29	88.04	93.97	89.63	87.36	92.07	86.95	87.61	90.57
GMCF	96.52	91.79	90.40	91.68	90.70	87.94	91.30	88.45	87.50	90.69
GCCF	98.02	94.34	91.53	92.70	90.86	86.61	92.57	88.66	88.38	91.51

Methods	Purity (%)									
	4	6	8	10	12	14	16	18	20	Avg.
Kmeans	73.84	79.08	88.41	75.91	78.11	67.52	73.66	68.85	66.41	74.64
NMF	80.83	79.74	86.25	73.31	71.25	70.67	72.89	68.88	67.35	74.57
CF	81.36	78.73	82.57	74.34	69.78	70.37	69.93	63.56	62.22	72.54
LCCF	88.76	85.12	92.36	83.60	85.05	74.68	79.09	77.24	73.61	82.16
GCF	97.73	89.67	87.83	85.22	78.12	73.84	81.70	68.70	69.73	81.39
ADGCF	97.52	92.93	88.58	93.55	91.66	90.47	95.13	85.95	86.59	91.37
GMCF	96.52	91.97	90.31	93.33	90.04	88.69	91.49	87.57	89.51	91.04
GCCF	98.94	97.89	95.05	92.70	93.31	90.97	92.26	87.58	88.25	92.99

Table 6: Clustering result on AR

Methods	Accuracy (%)								
	3	4	5	6	7	8	9	10	Avg.
Kmeans	58.97	45.54	40.30	26.92	26.37	30.28	33.33	23.07	36.79
NMF	52.56	46.15	40.07	26.28	36.81	30.25	28.20	26.92	42.58
CF	61.53	41.34	40.84	31.41	24.17	27.88	25.64	29.23	38.86
LCCF	52.56	48.07	45.38	38.46	37.36	29.32	29.91	22.30	37.98
GCF	57.69	44.23	42.30	35.25	32.96	29.80	33.38	24.61	39.58
ADGCF	62.82	53.84	56.15	30.76	29.67	28.84	29.91	26.92	44.79
GMCF	56.41	51.92	43.07	32.05	37.36	31.73	34.18	26.15	43.48
GCCF	73.07	61.53	58.46	43.59	41.75	37.54	34.16	30.76	47.61

Methods	Normalized Mutual Information (%)								
	3	4	5	6	7	8	9	10	Avg.
Kmeans	40.86	39.67	23.03	15.08	18.66	31.23	33.03	23.02	28.07
NMF	24.25	36.68	25.11	14.94	29.39	27.57	27.95	32.62	27.31
CF	25.54	15.44	20.95	17.64	13.17	23.13	23.08	25.38	20.54
LCCF	27.74	41.98	38.99	26.75	34.36	24.60	26.19	20.65	30.15
GCF	29.31	25.85	25.61	25.95	28.57	29.80	29.44	26.94	27.68
ADGCF	42.58	47.63	48.87	21.18	23.91	21.26	26.20	26.24	32.23
GMCF	35.13	34.40	24.84	18.66	30.65	33.35	33.58	27.87	29.81
GCCF	44.20	50.35	48.53	34.91	38.11	37.61	29.07	32.98	39.34

Methods	Purity (%)								
	3	4	5	6	7	8	9	10	Avg.
Kmeans	70.92	67.30	51.69	44.87	46.15	49.51	50.42	37.69	52.31
NMF	62.82	65.38	50.69	42.94	48.35	47.59	46.15	39.23	50.39
CF	66.66	59.61	48.46	45.51	36.26	46.15	50.40	45.38	49.80
LCCF	62.82	69.23	57.69	44.87	50.54	48.55	44.44	39.23	52.17
GCF	71.79	65.38	52.30	46.79	48.35	50.33	39.31	40.76	51.87
ADGCF	75.64	72.92	61.38	53.61	47.25	52.88	39.38	42.30	55.67
GMCF	65.38	59.61	47.69	42.94	49.45	47.11	41.88	39.23	49.16
GCCF	77.07	75.00	63.84	51.92	51.64	49.15	46.59	43.07	57.28

Table 7: Clustering result on CMU PIE

Methods	Accuracy (%)							
	10	20	30	40	50	60	68	Avg.
Kmeans	31.19	28.45	26.58	25.23	25.42	23.79	23.59	26.32
NMF	49.76	54.64	53.61	53.06	51.66	48.98	47.54	51.32
CF	37.61	24.88	22.38	22.44	13.42	12.71	11.42	36.60
LCCF	41.66	37.02	33.17	29.91	27.50	25.69	25.49	31.49
GCF	29.52	21.66	14.36	14.28	17.11	17.65	15.35	18.56
ADGCF	78.09	62.67	63.57	61.36	60.40	58.57	54.07	62.67
GMCF	68.57	60.95	60.43	60.08	59.21	61.46	55.87	60.93
GCCF	78.69	76.98	75.41	73.92	72.47	73.43	72.65	74.79
Methods	Normalized Mutual Information (%)							
	10	20	30	40	50	60	68	Avg.
Kmeans	35.67	47.65	50.26	49.34	52.36	52.57	53.16	48.71
NMF	59.43	72.39	74.53	76.21	76.87	77.91	75.48	73.26
CF	42.94	35.66	35.75	43.81	28.94	31.12	30.39	35.51
LCCF	58.95	62.80	58.41	59.41	58.27	58.23	57.38	59.06
GCF	34.87	35.21	28.26	30.08	38.82	40.33	39.09	35.23
ADGCF	84.53	81.07	83.53	84.07	83.81	83.29	80.58	82.98
GMCF	80.43	78.79	82.52	82.92	82.48	84.02	81.59	81.82
GCCF	85.46	89.71	85.77	87.39	87.51	87.99	85.86	87.09
Methods	Purity (%)							
	10	20	30	40	50	60	68	Avg.
Kmeans	36.42	37.38	35.95	31.96	31.64	30.69	30.53	33.51
NMF	55.47	62.54	62.54	61.04	59.66	58.94	56.24	59.49
CF	46.66	31.07	30.79	31.54	20.04	17.77	17.09	27.85
LCCF	52.61	47.97	43.81	41.75	37.73	34.14	33.24	41.60
GCF	44.28	30.95	20.39	21.90	22.28	23.98	22.62	26.62
ADGCF	87.38	79.58	78.73	80.26	77.14	78.96	80.06	80.30
GMCF	82.59	72.79	77.18	81.25	77.21	79.60	81.81	78.91
GCCF	88.69	89.84	84.58	85.65	85.31	87.36	84.51	86.56

Figure 3: Clustering performance of the eight methods versus parameter p on five dataset-s. Each row denotes the clustering performance of a dataset. From up to down, the rows correspond to TDT2, Reuters, COIL20, AR, and CMU PIE respectively.

Figure 4: Clustering performance of the eight methods versus parameter λ on five dataset-s. Each row denotes the clustering performance of a dataset. From up to down, the rows correspond to TDT2, Reuters, COIL20, AR, and CMU PIE respectively.

Figure 5: Convergence curves of GCCF : (a) TDT2; (b) Reuters; (c) COIL20; (d) AR; (e) CMU PIE.

Table 8: Average Computational Time (sec) on five datasets

Methods	TDT2	Reuters	COIL20	AR	CMU PIE
Kmeans	0.68	0.06	0.01	0.008	0.009
NMF	16.95	0.84	0.11	0.07	0.14
CF	15.05	0.12	0.14	0.08	0.12
LCCF	24.23	3.54	0.16	0.10	0.13
GCF	27.60	5.87	0.22	0.12	0.16
ADGCF	2102.26	259.82	2.15	1.01	1.61
GMCF	153.61	19.98	0.45	0.14	0.21
GCCF	1304.29	172.82	1.61	0.59	1.16

6. Conclusion

In this paper, we have proposed a robust graph regularized concept
 335 factorization method called correntropy based graph regularized concept
 factorization (GCCF). GCCF directly incorporates the graph structure
 information into the proposed objective function which is based on the
 maximum correntropy criterion (MCC). As a result, GCCF is insensitive to
 non-Gaussian noises and outliers in the data. Using the half-quadratic
 340 optimization technique, we solve effectively the non-convex objective function
 of GCCF, and derive the multiplicative update rules. In addition, we analyze
 the proposed GCCF algorithm. Experimental results show that the GCCF
 method can achieve good performance in clustering tasks on different types of
 real world datasets.

345 There are several issues in the proposed GCCF method to be investigated
in the future work. First, the parameters K , p , and λ have great influence on
the overall performance of GCCF. However, up to now, there is no systematic
strategy to select these parameters. Therefore, how to select the optimal
parameters is a challenging work. Second, GCCF suffers from the high
350 computational complexity. In the future, investigation will be carried out to
develop a more computationally efficient method for GCCF.

Acknowledgments

This work was partially supported by Nanyang Technological University
Research Scholarships. The work described in this paper was also partially
355 supported by 973 Program (No. 2015CB351703), National Natural Science
Foundation of China (No. 91648208), and the National Natural Science
Foundation-Shenzhen Joint Research Program (U1613219).

References

- [1] H. Kriegel, P. Krger, A. Zimek, Clustering high-dimensional data: A sur-
360 vey on subspace clustering, pattern-based clustering, and correlation clus-
tering, ACM Transactions on Knowledge Discovery from Data (TKDD)
3 (1) (2009) 1–58.
- [2] Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data
representation, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015)
365 2085–2098.
- [3] Y. Song, Q. Li, H. Huang, D. Feng, M. Chen, W. Cai, Low Dimensional
Representation of Fisher Vectors for Microscopy Image Classification, IEEE
Transactions on Medical Imaging 36 (8) (2017) 1636–1649.
- [4] W. Chen, Y. Zhao, B. Pan, B. Chen, Supervised kernel nonnegative matrix
370 factorization for face recognition, Neurocomputing 205 (2016) 165–181.

- [5] D. Tao, X. Li, X. Wu, S. Maybank, Geometric mean for subspace selection, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2) (2009) 260–274.
- [6] G. Cui, X. Li, Y. Dong, Subspace clustering guided convex nonnegative matrix factorization, *Neurocomputing* 292 (2018) 38–48.
- 375 [7] H. Li, J. Zhang, G. Shi, J. Liu, Graph-based discriminative nonnegative matrix factorization with label information, *Neurocomputing* 266 (2017) 91–100.
- [8] A. Gersho, R. Gray, *Vector Quantization and Signal Compression*, Boston, MA, USA: Kluwer 1992.
- 380 [9] R. Duda, P. Hart, D. Stork, *Pattern Classification*, Hoboken, NJ, USA: Wiley-Interscience 2000.
- [10] I. Jolliffe, *Principal Component Analysis*, New York, NY, USA: Springer-Verlag 1989.
- [11] B. Jiang, C. Ding, B. Luo, Robust data representation using locally linear embedding guided PCA, *Neurocomputing* 275 (2018) 523–532.
- 385 [12] D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* 401 (6755) (1999) 788–791.
- [13] R. Sandler, M. Lindenbaum, Nonnegative matrix factorization with earth mover’s distance metric for image analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (8) (2011) 1590–1602.
- 390 [14] W. Xu, Y. Gong, Document clustering by concept factorization, *Proc. ACM Int. Conf. Res. Develop. Inform. Retrieval* (2004) 202–209.
- [15] H. Liu, G. Yang, Z. Wu, D. Cai, Constrained concept factorization for image representation, *IEEE Transactions on cybernetics* 44 (7) (2014) 1214–1224.
- 395 [16] D. Lee, H. Seung, Algorithms for non-negative matrix factorization, *Advances in neural information processing systems* (2001) 556–562.

- [17] D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (8) (2011) 1548–1560.
- 400 [18] X. Li, G. Cui, Y. Dong, Graph regularized non-negative low-rank matrix factorization for image clustering, *IEEE transactions on cybernetics* 47 (11) (2017) 3840–3853.
- [19] H. Gao, F. Nie, H. Huang, Local Centroids Structured Non-negative Matrix Factorization, *AAAI* (2017) 1905–1911.
- 405 [20] P. Li, C. Chen, J. Bu, Clustering analysis using manifold kernel concept factorization, *Neurocomputing* 87 (15) (2012) 120–131.
- [21] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, *IEEE Transactions on Knowledge and Data Engineering* 23 (6) (2011) 902–913.
- 410 [22] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, *Neurocomputing* 138 (22) (2014) 120–130.
- [23] H. Liu, Z. Yang, J. Yang, Z. Wu, X. Li, Local coordinate concept factorization for image representation, *IEEE Transactions on Neural Networks and Learning Systems* 25 (6) (2014) 1071–1082.
- 415 [24] C. Peng, Z. Kang, Y. Hu, J. Cheng, Q. Cheng, Robust Graph Regularized Nonnegative Matrix Factorization for Clustering, *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11 (3).
- [25] R. He, B. Hu, W. Zheng, X. Kong, Robust principal component analysis based on maximum correntropy criterion, *IEEE Transactions on Image Processing* 20 (6) (2011) 1485–1494.
- 420 [26] J. Principe, *Information Theoretic Learning: Renyi’s Entropy and Kernel Perspectives*, Springer: New York, NY, USA, 2010.

- [27] R. He, W. Zheng, B. Hu, Maximum correntropy criterion for robust face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (8) (2011) 1561–1576.
- [28] E. Hasanbelliu, L. Giraldo, J. Principe, Information theoretic shape matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (12) (2014) 2436–2451.
- [29] Y. Wang, S. Wu, M. B. X. Zhang, Z. Luo, Correntropy induced metric based graph regularized non-negative matrix factorization, *Neurocomputing* 204 (5) (2016) 172–182.
- [30] J. Wang, X. Wang, X. Gao, Non-negative matrix factorization by maximizing correntropy for cancer clustering, *BMC bioinformatics* 14 (1) (2013) 107–118.
- [31] S. Peng, B. Chen, S. Lei, W. Ser, Z. Lin, Constrained maximum correntropy adaptive filtering, *Signal Processing* 140 (2017) 116–126.
- [32] B. Chen, Y. Zhu, J. Hu, J. Principe, *System Parameter Identification: Information Criteria and Algorithms*, Newnes, 2013.
- [33] N. Zhou, Y. Xu, H. Cheng, Z. Yuan, B. Chen, Maximum Correntropy Criterion based Sparse Subspace Learning for Unsupervised Feature Selection, *IEEE Transactions on Circuits and Systems for Video Technology* PP (99) (2017) 1–13.
- [34] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, Z. Lin, Correntropy induced l2 graph for robust subspace clustering, *Proceedings of the IEEE International Conference on Computer Vision* (2013) 1801–1808.
- [35] W. Liu, P. Pokharel, J. Principe, Correntropy: Properties and applications in non-gaussian signal processing, *IEEE Transactions on Signal Processing* 55 (11) (2007) 5286–5298.
- [36] R. He, W. Zheng, T. Tan, Z. Sun, Half-quadratic-based iterative minimization for robust sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (2) (2014) 261–275.

- [37] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, *Advances in neural information processing systems* (2002) 585–591.
- [38] L. Weng, F. Dornaika, Z. Jin, Graph construction based on data self-representativeness and laplacian smoothness, *Neurocomputing* 207 (2016) 476–487.
- [39] B. Chen, J. Wang, H. Zhao, N. Zheng, J. Principe, Convergence of a fixed-point algorithm under maximum correntropy criterion, *IEEE Signal Processing Letters* 22 (10) (2015) 1723–1727.
- [40] B. Chen, J. Principe, Maximum correntropy estimation is a smoothed MAP estimation, *IEEE Signal Processing Letters* 19 (8) (2012) 491–494.
- [41] S. Boyd, *Convex Optimization*, Cambridge university press 2004.
- [42] D. Bertsekas, *Nonlinear Programming*, Belmont: Athena scientific 1999.
- [43] X. Huang, X. Lou, B. Cui, A novel neural network for solving convex quadratic programming problems subject to equality and inequality constraints, *Neurocomputing* 214 (2016) 23–31.
- [44] Y. Wang, C. Pan, S. Xiang, F. Zhu, Robust hyperspectral unmixing with correntropy-based metric, *IEEE Transactions on Image Processing* 24 (11) (2015) 4027–4040.
- [45] P. Hoyer, Non-negative sparse coding, *Neural Networks for Signal Processing* (2002) 557–565.
- [46] T. Moon, The expectation-maximization algorithm, *IEEE Signal processing magazine* 13 (6) (1996) 47–60.
- [47] C. Lin, Projected gradient methods for nonnegative matrix factorization, *Neural computation* 19 (10) (2007) 2756–2779.

- [48] F. Sha, Y. Lin, L. Saul, D. Lee, Multiplicative updates for nonnegative quadratic programming, *Neural computation* 19 (8) (2007) 2004–2031.
- [49] A. Martinez, R. Benavente, The AR Face Database, Computer Vision Center technical report.
- 480 [50] L. Lovasz, M. Plummer, *Matching Theory*, North Holland 1986.
- [51] J. Ye, Z. Jin, Feature selection for adaptive dual-graph regularized concept factorization for data representation, *Neural Processing Letters* 45 (2) (2017) 667–688.
- [52] X. Li, X. Shen, Z. Shu, Q. Ye, C. Zhao, Graph regularized multilayer
485 concept factorization for data representation, *Neurocomputing* 238 (2017) 139–151.