Robust Non-negative Graph Embedding: Towards Noisy Data, Unreliable Graphs, and Noisy Labels

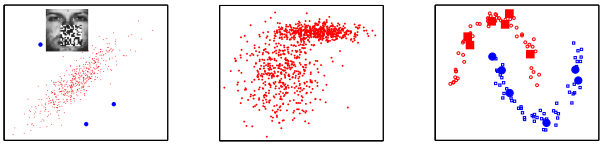
Hanwang Zhang[†], Zheng-Jun Zha[†], Shuicheng Yan[‡], Meng Wang[†], Tat-Seng Chua[†]
[†]School of Computing, National University of Singapore
[‡]Electrical Computer Engineering, National University of Singapore
{hanwang,zhazj,wangm,chuats}@comp.nus.edu.sg;eleyans@nus.edu.sg

Abstract

Non-negative data factorization has been widely used recently. However, existing techniques, such as Non-negative Graph Embedding (NGE), often suffer from noisy data, unreliable graphs, and noisy labels, which are commonly encountered in real-world applications. To address these issues, in this paper, we propose a Robust Non-negative Graph Embedding (RNGE) framework. The joint sparsity in both graph embedding and reconstruction endues the robustness of RNGE. We develop an elegant multiplicative updating solution that can solve RNGE efficiently and prove the convergence rigourously. RNGE is robust to unreliable graphs, as well as both sample and label noises in training data. Moreover, RNGE provides a general formulation such that all the algorithms unified with the graph embedding framework can be easily extended to obtain their robust non-negative solutions. We conduct extensive experiments on four real-world datasets and compared the proposed RNGE to NGE and other representative non-negative data factorization and subspace learning methods. The experimental results demonstrate the effectiveness and robustness of RNGE.

1. Introduction

Inspired by the psychological and physiological evidence for part-based representations in human vision system [6], the techniques of nonnegative data factorization have attracted increasing attentions recently. These techniques seek for the sparse nonnegative bases for data representation, where a sample can be formed as a non-subtractive combination of the bases. As a pioneering work, Nonnegative Matrix Factorization (NMF) [15] triggers many subsequent studies [16][8][10]. Most of these algorithms are unsupervised and motivated for data reconstruction, and hence lack discriminative power for classification. Recently, Yang et al. [25] proposed a unified non-negative data factorization framework, called Nonnegative Graph Embedding (NGE). Besides the nonnegative reconstruction function in NMF, a graph embedding function is exploited in NGE. The ultimate data factorization is separated into two parts, which separatively preserve the similarities measured by the intrinsic and penalty graphs [24], and together minimize the data reconstruction error. NGE is applicable for supervised/semi-supervised configuration, and hence possesses classification capability. Despite the appealing properties of NGE, it suffers from the following problems in real-world applications.



(a) Noisy Data (b) Uneven Distribution (c) Noisy Labels

Figure 1. Illustrations of noisy data, uneven data distribution, and noisy labels. Fig.(a) illustrates several noisy samples, for example, the face images corrupted by occlusions. Fig.(b) shows the uneven distribution, where the data distribution varies greatly at different areas in the feature space. Fig.(c) illustrates two-moons toy data with each class having several incorrectly labeled data points, which are highlighted.

- **Noisy Data** In general, data may consist of undesirable noises and outliers due to occlusions (e.g. a hand in front of a face), image noises (e.g. from scanning archival data), or illuminations (e.g. specular reflections), etc (see Fig. 1(a)). Although NGE can deal with noises in testing data to some extend, it suffers from significant performance degradation when noisy training samples come in, since even a single outlier or noisy datum can dominate the sum of square errors in the reconstruction functions of NGE.

- **Unreliable Graphs** The two existing popular ways for constructing the intrinsic and penalty graphs in NGE

are k-nearest neighbor and ϵ-ball based methods. For a sample, its k-nearest neighbors or the samples within its surrounding ϵ-ball are connected. The edge weights can then be set by various strategies, such as binary, Heat kernel and Gaussian Kernel, etc. However, such graph becomes unreliable when unfavorable noisy data come in. Furthermore, as illustrated in Fig. 1(b), the data distribution may be uneven and vary greatly at different areas in the feature space. This results in distinctive local structure for each datum. Both k-nearest neighbor and ϵ-ball based methods adopt a fixed global parameter to determine the neighbors of all the data. Hence, the resultant graph cannot precisely characterize the desired local structure of the sample space.

- **Noisy Labels** One challenge in real-word classification tasks is the lack of expert-labeled training data, while the data labeled by amateur annotators have been often exploited as an alternative. Unfortunately, some training samples might be labeled with incorrect classes (Fig. 1(c)). The noisy labels may severely degrade the performance of NGE.

To address all the above issues, in this paper, we propose a novel Robust Nonnegative Graph Embedding (RNGE) framework, where both the graph embedding and data reconstruction functions are formulated in ℓ_1-norm manner. With the the joint sparsity in graph embedding and data reconstruction induced by ℓ_1-norm, RNGE possesses robustness to noisy data, unreliable graphs, as well as noisy labels. While the ℓ_1-norm reconstruction function alleviates the influence of noisy data, the ℓ_1-norm embedding functions lead to sparse and more ideal embedding results, and hence endow the robustness of RNGE to unreliable graphs and noisy labels. Also, RNGE provides a general framework such that all the algorithms, both supervised and unsupervised, unified with the graph embedding framework can be easily extended to obtain their robust non-negative solutions. We derive the computational algorithm and provide rigorous analysis on its convergence. We show that the derived solution for RNGE has elegant multiplicative updating rules and is easy for implementation. Experimental results on four real-world datasets show that our RNGE consistently outperforms NGE in terms of classification results, and possess robustness to noisy data, unreliable graphs, and noisy labels.

The remainder of this paper is organized as follows. We review related works in Section 2, and briefly introduce the NGE algorithm in Section 3. Our proposed Robust NGE is elaborated in Section 4. Section 5 demonstrates the detailed experimental results, followed by the conclusions in Section 6.

2. Related Works

Recently, ℓ_1-norm is successfully used in sparse representation [23][11] and robust low-rank data factorization [3][13]. For example, Ke et al. [12] proposed a unified ℓ_1-norm subspace learning framework based on linear programming technique, however it is time consuming. Ding et al. [5] used $\ell_{2,1}$-norm (R_1-norm) instead of ℓ_2-norm to achieve robust PCA. Nojun [14] proposed a robust ℓ_1-norm PCA motivated from maximizing the dual problem of low-rank decomposition. Kong et al. [13] and used $\ell_{2,1}$-norm instead of ℓ_2-norm in NMF reconstruction function to make NMF robust to outliers. Those studies positively show that ℓ_1-norm can suppress noises to some extend. To overcome unreliable graphs problem, Pang and Yuan [19] *maximized* an ℓ_1-norm dissimilarity graph cost function to improve the robustness of the locality preserving projection (LPP) [9]. Lately, Nie et al. [18] proposed an ℓ_1-norm graph clustering method which adaptively re-weights the original weights of graph to discover clearer cluster structure.

In addition, there exist several works aiming at improving the original NGE [25]. Wang et al. [21] derived an efficient algorithm for solving NGE, instead of calculating the inverse of the so-called M-matrix in each iteration, and hence significantly reduced the time complexity. Cai et al. [2] and Gu et al. [7] proposed multiplicative updating rules for unsupervised NGE. Liu et al. [17] proposed a projective NGE approach addressing the out-of-sample issue in classification, yet it is a LPP implementation of NGE. These achieve improvements on computational speed or generalization ability, but lack robustness.

3. NGE Revisited

In this section, we briefly review the Non-negative Graph Embedding (NGE) approach, starting with some notations. We use a boldface lowercase letter $\mathbf{v}$ to denote a vector, and use a boldface uppercase letter $\mathbf{M}$ to denote a matrix. $\mathbf{M}_{ij}$ is the entry at the ith row and jth column. $\mathbf{M}_{i\cdot}$ and $\mathbf{M}_{\cdot j}$ denote the vectors composed by the ith row and the jth column of $\mathbf{M}$, respectively. The function $\|\cdot\|$ denotes the norm of a matrix (or a vector). Particularly, $\|\cdot\|_F$ is the Frobenius norm for a matrix, $\|\cdot\|_2$ is the ℓ_2-norm for a vector and $\|\cdot\|_1$ is the ℓ_1-norm which is the sum of the absolute value of all the entries in the matrix (or vector). The operator $\odot$ denotes the Hadamard entry-wise matrix multiplication.

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ denote a set of training samples, where $\mathbf{x}_i \in \mathbb{R}^m$ and N is the total number of training samples. The corresponding class labels are denoted as $\{c_i | c_i \in \{1, \cdots, N_c\}\}_{i=1}^N$ with N_c is the class number. As shown in Eq. (1) below, NGE separates the ultimate data decomposition into two parts, which separatively preserve the similarities measured by the intrinsic and penalty graphs,

and meanwhile minimizes the data reconstruction errors. Specially, in supervised setting, the intrinsic graph characterizes the intra-class compactness and connects each data point with its neighboring points of the same class, while the penalty graph connects the marginal points and characterizes the inter-class separability.

$$\begin{aligned}\min_{\mathbf{V},\mathbf{U}} \ & \sum_{i,j}\left\|\mathbf{Q}^1(\mathbf{V}^1_{\cdot i}-\mathbf{V}^1_{\cdot j})\right\|_2^2\mathbf{S}_{ij}+\sum_{i,j}\left\|\mathbf{Q}^2(\mathbf{V}^2_{\cdot i}-\mathbf{V}^2_{\cdot j})\right\|_2^2\mathbf{S}^p_{ij} \\ & +\lambda\|\mathbf{X}-\mathbf{U}\mathbf{V}\|_F^2, \quad s.t.\ \mathbf{U},\mathbf{V}\geq\mathbf{0},\end{aligned} \tag{1}$$

where $\mathbf{S}_{ij}$ and $\mathbf{S}^p_{ij}$ are the edge weights of the intrinsic and penalty graphs respectively. Typically, they can be set as:

$$\mathbf{S}_{ij}=\begin{cases}1, & i\in N_{k_1}(j)\ or\ j\in N_{k_1}(i)\\ 0 & otherwise\end{cases} \tag{2}$$

where $N_{k_1}(i)$ indicates the index set of the k_1 nearest neighbors of $\mathbf{x}_i$ in the *same* class, and

$$\mathbf{S}^p_{ij}=\begin{cases}1 & i\in P_{k_2}(j)\ or\ j\in P_{k_2}(i)\\ 0 & otherwise\end{cases} \tag{3}$$

where $P_{k_2}(i)$ indicates the index set of the k_2 nearest neighbors of $\mathbf{x}_i$ in the *distinct* classes. Specially, we can also define the weights in an unsupervised way, i.e., $\mathbf{S}_{ij}=1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the k-nearest neighborhood and 0 otherwise. Then, the second term of the penalty embedding in Eq. (1) will vanish. $\mathbf{V}\in\mathbb{R}^{r\times N}(r\ll m)$ is the reconstruction coefficient matrix and $\mathbf{U}\in\mathbb{R}^{m\times r}$ is the matrix of bases. $\mathbf{V}$ is divided into two parts as $\mathbf{V}=[\mathbf{V}^1;\ \mathbf{V}^2]$ with $\mathbf{V}^1\in\mathbb{R}^{d\times N}(d<r)$ and $\mathbf{V}^2\in\mathbb{R}^{(r-d)\times N}$. Correspondingly, $\mathbf{U}$ is separated as $[\mathbf{U}^1\ \ \mathbf{U}^2]$. $\mathbf{Q}^1=diag(\|\mathbf{U}^1_{\cdot 1}\|,\|\mathbf{U}^1_{\cdot 2}\|,\cdots,\|\mathbf{U}^1_{\cdot d}\|)$ and $\mathbf{Q}^2=diag(\|\mathbf{U}^2_{\cdot 1}\|,\|\mathbf{U}^2_{\cdot 2}\|,\cdots,\|\mathbf{U}^2_{\cdot r-d}\|)$ are used to prevent $\mathbf{V}$ from arbitrarily down-scaling. The first term in Eq. (1) aims to retain intrinsic graph properties, which maps neighbors in the original space as neighbors (or even the same point) in the low-dimensional space $\mathbf{V}^1$. Meanwhile, in order to maximize the distance of the non-neighbors in the penalty graph, a complementary low-dimensional embedding $\mathbf{V}^2$ in the second term is constructed to harmonize the two opposite criteria into a single "minimization" one. Then, $(\mathbf{U}^1,\mathbf{V}^1)$ and $(\mathbf{U}^2,\mathbf{V}^2)$ together reconstruct the original data in an additive manner (in the third term). NGE is a general framework such that all the algorithms unified within the graph embedding framework can be extended to obtain their non-negative solutions.

4. Robust Non-negative Graph Embedding

As mentioned above, NGE is sensitive to data noises, unreliable graphs and noisy labels, which are commonly encountered in real-world applications. In order to tackle these issues, we propose a Robust Non-negative Graph Embedding (RNGE) framework in this section.

4.1. Formulation

Recall the objective function of NGE in Eq. (1). First, the reconstruction errors on samples are squared. Thus a few noisy samples with large errors may easily dominate the objective function. Second, the intrinsic and penalty graphs play important roles in NGE. Hence, NGE suffers from significant performance degradation when the graphs have undesirable structure. For example, when two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are mislinked as neighbors in the graph, i.e., $\mathbf{S}_{i,j}\neq 0$ or $\mathbf{S}^p_{i,j}\neq 0$, the embedding term $\|\mathbf{Q}^1(\mathbf{V}^1_{\cdot i}-\mathbf{V}^1_{\cdot j})\|_2^2\mathbf{S}_{ij}$ or $\|\mathbf{Q}^2(\mathbf{V}^2_{\cdot i}-\mathbf{V}^2_{\cdot j})\|_2^2\mathbf{S}^p_{ij}$ may bring undesirable noises into the objective function and negatively impact the optimization results. Third, the intrinsic graph is constructed over the samples from the same classes, while the penalty graph links the samples from distinctive classes. Thus, when the noisy labels come in, incorrect links emerge and result in performance degradation.

Towards robustness to noisy data, unreliable graphs, and noisy labels, we propose a Robust NGE formulated based on ℓ_1-norm as follows:

$$\begin{aligned}\min_{\mathbf{V},\mathbf{U}} \ & \sum_{i,j}\left\|\mathbf{Q}^1(\mathbf{V}^1_{\cdot i}-\mathbf{V}^1_{\cdot j})\right\|_2\mathbf{S}_{ij}+\sum_{i,j}\left\|\mathbf{Q}^2(\mathbf{V}^2_{\cdot i}-\mathbf{V}^2_{\cdot j})\right\|_2\mathbf{S}^p_{ij} \\ & +\lambda\|\mathbf{X}-\mathbf{U}\mathbf{V}\|_1, \quad s.t.\ \mathbf{U},\mathbf{V}\geq\mathbf{0}\end{aligned}, \tag{4}$$

where the reconstruction term $\|\mathbf{X}-\mathbf{U}\mathbf{V}\|_1$ suppresses both outliers and corrupted samples that contain noises on certain elements. The large errors from outliers and noises do not dominate the objective function, since they are not squared. Similarly, the embedding terms alleviate the influence of incorrect $\mathbf{S}_{ij}$ or $\mathbf{S}^p_{ij}$ in unreliable graphs caused by noisy data, uneven distribution, or noisy labels, since the embedding errors on sample pairs are not squared. More interestingly, the embedding term $\sum_{i,j}\left\|\mathbf{Q}^1(\mathbf{V}^1_{\cdot i}-\mathbf{V}^1_{\cdot j})\right\|_2\mathbf{S}_{ij}$ can be considered as the ℓ_1-norm of a sample-pair vector $\mathbf{p}$ whose $(N\times i+j)$-th element is $\|\mathbf{Q}^1(\mathbf{V}^1_{\cdot i}-\mathbf{V}^1_{\cdot j})\|_2\mathbf{S}_{ij}$, while $\sum_{i,j}\left\|\mathbf{Q}^2(\mathbf{V}^2_{\cdot i}-\mathbf{V}^2_{\cdot j})\right\|_2\mathbf{S}^p_{ij}$ is the ℓ_1-norm of a sample-pair vector $\mathbf{q}$ with elements $\|\mathbf{Q}^2(\mathbf{V}^2_{\cdot i}-\mathbf{V}^2_{\cdot j})\|_2\mathbf{S}^p_{ij}$. Note that the minimization of ℓ_1-norm leads to sparse solutions [23]. That is to say, many elements in $\mathbf{p}$ and $\mathbf{q}$ are zeros, i.e., $\mathbf{V}^1_{\cdot i}=\mathbf{V}^1_{\cdot j}$ and $\mathbf{V}^2_{\cdot i}=\mathbf{V}^2_{\cdot j}$ for many sample pairs. Recall that $\mathbf{V}^1_{\cdot i}$ is an embedded point in the "intrinsic" space and is embedded by the intrinsic graph that links samples within the same classes. The sparse solution of $\mathbf{p}$ provides more compact intra-class embedding than NGE. Recall that $\mathbf{V}^2_{\cdot i}$ is in the complementary space and is embedded by the penalty graph connects samples from distinct classes. The sparse solution of $\mathbf{q}$ leads to more compact embedding in the complementary space. This essentially boosts the inter-class separability and intra-class compactness, since neighbors in the complementary space implies non-neighbors in the counterpart.

Note that the objective function of RNGE in Eq. (4) is non-smooth and difficult to be solved efficiently, and the

multiplicative updating solution for NGE is not applicable here. In the next subsections, we develop a new set of multiplicative updating rules to solve RNGE efficiently and provide rigorous analysis on the convergence.

4.2. Solution

We solve the optimization problem in Eq. (4) by iteratively solving its two sub-problems: optimizing $\mathbf{U}$ with a fixed $\mathbf{V}$ and optimizing $\mathbf{V}$ with a fixed $\mathbf{U}$. To tackle the nonsmoothness, we reformulate each sub-problem into an equivalent ℓ_2-norm form.

Optimize U for Given V: For a fixed $\mathbf{V}$, we re-weight the ℓ_1-norm objective function in Eq. (4) by the latest $\mathbf{U}^{(t)}$ into an ℓ_2-norm form as follows:

$$\begin{aligned} F(\mathbf{U}) = &\sum_{i,j} \left\| \mathbf{Q}^1 \mathbf{V}^1_{\cdot i} - \mathbf{Q}^1 \mathbf{V}^1_{\cdot j} \right\|_2^2 \frac{\mathbf{S}_{ij}}{\left\| \mathbf{Q}^{1(t)} \mathbf{V}^1_{\cdot i} - \mathbf{Q}^{1(t)} \mathbf{V}^1_{\cdot j} \right\|_2} \\ &+ \sum_{i,j} \left\| \mathbf{Q}^2 \mathbf{V}^2_{\cdot i} - \mathbf{Q}^2 \mathbf{V}^2_{\cdot j} \right\|_2^2 \frac{\mathbf{S}^p_{ij}}{\left\| \mathbf{Q}^{2(t)} \mathbf{V}^2_{\cdot i} - \mathbf{Q}^{2(t)} \mathbf{V}^2_{\cdot j} \right\|_2} \\ &+ \sum_{i,j} \left| \mathbf{X}_{ij} - (\mathbf{U}\mathbf{V})_{ij} \right|^2 \frac{\lambda}{\left| \mathbf{X}_{ij} - (\mathbf{U}^{(t)}\mathbf{V})_{ij} \right|} \end{aligned} \tag{5}$$

Let $\tilde{\mathbf{S}}_{ij} = \frac{\mathbf{S}_{ij}}{\|\mathbf{Q}^{1(t)} \mathbf{V}^1_{\cdot i} - \mathbf{Q}^{1(t)} \mathbf{V}^1_{\cdot j}\|_2}$, $\tilde{\mathbf{S}}^p_{ij} = \frac{\mathbf{S}^p_{ij}}{\|\mathbf{Q}^{2(t)} \mathbf{V}^2_{\cdot i} - \mathbf{Q}^{2(t)} \mathbf{V}^2_{\cdot j}\|_2}$, we can rewrite Eq. (5) in a more compact form:

$$F(\mathbf{U}) = Tr(\mathbf{U}\tilde{Y}_u\mathbf{U}^T) + Tr\left((\mathbf{\Lambda} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}))^T (\mathbf{X} - \mathbf{U}\mathbf{V}) \right) \tag{6}$$

where

$$\begin{cases} \tilde{\mathbf{Y}}_u = \begin{bmatrix} \mathbf{V}^1 \tilde{\mathbf{L}} \mathbf{V}^{1T} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^2 \tilde{\mathbf{L}}^p \mathbf{V}^{2T} \end{bmatrix} \odot \mathbf{I} = \tilde{\mathbf{Y}}_{u^+} - \tilde{\mathbf{Y}}_{u^-}, \\ \tilde{\mathbf{Y}}_{u^+} = \begin{bmatrix} \mathbf{V}^1 \tilde{\mathbf{D}} \mathbf{V}^{1T} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^2 \tilde{\mathbf{D}}^p \mathbf{V}^{2T} \end{bmatrix} \odot \mathbf{I}, \\ \tilde{\mathbf{Y}}_{u^-} = \begin{bmatrix} \mathbf{V}^1 \tilde{\mathbf{S}} \mathbf{V}^{1T} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^2 \tilde{\mathbf{S}}^p \mathbf{V}^{2T} \end{bmatrix} \odot \mathbf{I}, \\ \mathbf{\Lambda}_{ij} = \lambda / |(\mathbf{X} - \mathbf{U}^{(t)}\mathbf{V})_{ij}|, \end{cases} \tag{7}$$

where $\tilde{\mathbf{D}} = diag(\sum_j \tilde{\mathbf{S}}_{ij})$, $\tilde{\mathbf{D}}^p = diag(\sum_j \tilde{\mathbf{S}}_{ij})$ and $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{S}}$, $\tilde{\mathbf{L}}^p = \tilde{\mathbf{D}}^p - \tilde{\mathbf{S}}^p$ are graph Laplacians. Note that Eq. (6) is quadratic. Thus by setting the derivatives of its Lagrange function to 0 [17], we can obtain the convergent multiplicative update rule for $\mathbf{U}^{(t+1)}$ as

$$\mathbf{U}^{(t+1)}_{ij} \leftarrow \mathbf{U}^{(t)}_{ij} \frac{\left((\mathbf{X} \odot \mathbf{\Lambda}) \mathbf{V}^T + \mathbf{U}^{(t)} \tilde{\mathbf{Y}}_{u-} \right)_{ij}}{\left(\left((\mathbf{U}^{(t)}\mathbf{V}) \odot \mathbf{\Lambda} \right) \mathbf{V}^T + \mathbf{U}^{(t)} \tilde{\mathbf{Y}}_{u+} \right)_{ij}} \tag{8}$$

It is obvious that $\mathbf{U}^{(t+1)}$ is non-negative if $\mathbf{V}$ and $\mathbf{U}^{(t)}$ are non-negative. We normalize the column vectors of $\mathbf{U}^{(t+1)}$ and consequently convey the norm to the coefficient matrix $\mathbf{V}$ as Eq. (9). The normalization will not change the primary objective function in Eq. (4).

$$\mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij} \times \|\mathbf{U}^{(t+1)}_{\cdot i}\|_2, \quad \mathbf{U}^{(t+1)}_{\cdot i} \leftarrow \mathbf{U}^{(t+1)}_{\cdot i} \big/ \|\mathbf{U}^{(t+1)}_{\cdot i}\|_2. \tag{9}$$

Optimize V for Given U: For a fixed and column-normalized $\mathbf{U}$, we re-weight the ℓ_1-norm objective function in Eq. (4) under $\mathbf{V}^{(t)}$ into an ℓ_2-norm form as:

$$\begin{aligned} F(\mathbf{V}) = &\sum_{i,j} \frac{\left\| \mathbf{V}^1_{i\cdot} - \mathbf{V}^1_{j\cdot} \right\|_2^2 \mathbf{S}_{ij}}{\left\| \mathbf{V}^{1(t)}_{i\cdot} - \mathbf{V}^{1(t)}_{j\cdot} \right\|_2} + \sum_{i,j} \frac{\left\| \mathbf{V}^2_{i\cdot} - \mathbf{V}^2_{j\cdot} \right\|_2^2 \mathbf{S}^p_{ij}}{\left\| \mathbf{V}^{2(t)}_{i\cdot} - \mathbf{V}^{2(t)}_{j\cdot} \right\|_2} \\ &+ \sum_{i,j} \left| \mathbf{X}_{ij} - (\mathbf{U}\mathbf{V})_{ij} \right|^2 \frac{\lambda}{\left| \mathbf{X}_{ij} - (\mathbf{U}\mathbf{V}^{(t)})_{ij} \right|} \end{aligned} \tag{10}$$

Redefine $\tilde{\mathbf{S}}_{ij} = \frac{\mathbf{S}_{ij}}{\|\mathbf{V}^{1(t)}_{i\cdot} - \mathbf{V}^{1(t)}_{j\cdot}\|_2}$, $\tilde{\mathbf{S}}^p_{ij} = \frac{\mathbf{S}^p_{ij}}{\|\mathbf{V}^{2(t)}_{i\cdot} - \mathbf{V}^{2(t)}_{j\cdot}\|_2}$, we can rewrite Eq. (10) in a more compact form as:

$$\begin{aligned} F(\mathbf{V}) = &Tr(\mathbf{V}^1 \tilde{\mathbf{Y}}^1_v \mathbf{V}^{1T}) + Tr(\mathbf{V}^2 \tilde{\mathbf{Y}}^2_v \mathbf{V}^{2T}) \\ &+ Tr\left((\mathbf{\Lambda} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}))^T (\mathbf{X} - \mathbf{U}\mathbf{V}) \right) \end{aligned} \tag{11}$$

where $\tilde{\mathbf{Y}}^1_v$ and $\tilde{\mathbf{Y}}^2_v$ are given as

$$\begin{cases} \tilde{\mathbf{Y}}^1_v = \tilde{\mathbf{Y}}^1_{v+} - \tilde{\mathbf{Y}}^1_{v-}, \tilde{\mathbf{Y}}^1_{v+} = \tilde{\mathbf{D}}, \tilde{\mathbf{Y}}^1_{v-} = \tilde{\mathbf{S}}, \\ \tilde{\mathbf{Y}}^2_v = \tilde{\mathbf{Y}}^2_{v+} - \tilde{\mathbf{Y}}^2_{v-}, \tilde{\mathbf{Y}}^2_{v+} = \tilde{\mathbf{D}}^p, \tilde{\mathbf{Y}}^2_{v-} = \tilde{\mathbf{S}}^p, \\ \mathbf{\Lambda}_{ij} = \lambda / |\mathbf{X}_{ij} - (\mathbf{U}\mathbf{V}^{(t)})_{ij}|. \end{cases} \tag{12}$$

Eq. (11) is quadratic. We set the derivatives of its Lagrange function as 0 [17], and obtain the multiplicative updating rule for $\mathbf{V}^{(t+1)}$ as:

$$\mathbf{V}^{(t+1)}_{ij} \leftarrow \mathbf{V}^{(t)}_{ij} \frac{\left(\mathbf{U}^T (\mathbf{X} \odot \mathbf{\Lambda}) + \begin{bmatrix} \mathbf{V}^{1(t)} \tilde{\mathbf{Y}}^1_{v-} \\ \mathbf{V}^{2(t)} \tilde{\mathbf{Y}}^2_{v-} \end{bmatrix} \right)_{ij}}{\left(\mathbf{U}^T ((\mathbf{U}\mathbf{V}^{(t)} \odot \mathbf{\Lambda})) + \begin{bmatrix} \mathbf{V}^{1(t)} \tilde{\mathbf{Y}}^1_{v+} \\ \mathbf{V}^{2(t)} \tilde{\mathbf{Y}}^2_{v+} \end{bmatrix} \right)_{ij}} \tag{13}$$

Note that $\mathbf{V}^{(t+1)}$ is nonnegative if $\mathbf{U}$ and $\mathbf{V}^{(t)}$ are nonnegative. With the resultant updating rules for $\mathbf{U}$ and $\mathbf{V}$, the optimization problem of RNGE in Eq. (4) can be solved by iteratively updating $\mathbf{U}$ and $\mathbf{V}$. The computation algorithm is in Algorithm 1. In the next subsection, we prove the convergence of Algorithm 1.

Algorithm 1 Robust Non-negative Graph Embedding

1: **Input:** The data matrix $\mathbf{X}$, the similarity matrix $\mathbf{S}$ and the penalty matrix $\mathbf{S}^p$.
2: **Output:** The non-negative basis matrix $\mathbf{U}$ and coefficient matrix $\mathbf{V}$.
3: **Initialization:** $t = 1$, set $\mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}$ as arbitrary non-negative matrices.
4: **Procedure:**
5: **while** *not converge* **do**
6: Calculate $\mathbf{\Lambda}$, $\tilde{\mathbf{Y}}_{u^+}$, $\tilde{\mathbf{Y}}_{u^-}$ according to Eq. (7).
7: Given $\mathbf{V} = \mathbf{V}^{(t)}$, update $\mathbf{U}^{(t+1)}$ according to Eq. (8).
8: Normalize $\mathbf{U}$ and scale $\mathbf{V}$ according to Eq. (9):
9: Set $\mathbf{V}^{(t)} = \mathbf{V}$ and calculate $\mathbf{\Lambda}$, $\tilde{\mathbf{Y}}^1_{v^+}$, $\tilde{\mathbf{Y}}^1_{v^-}$, $\tilde{\mathbf{Y}}^2_{v^+}$, $\tilde{\mathbf{Y}}^2_{v^-}$ according to Eq. (12).
10: Given $\mathbf{U} = \mathbf{U}^{(t)}$, update $\mathbf{V}^{(t+1)}$ according to Eq. (13).
11: $t = t + 1$.
12: **end while**
13: **Return** $\mathbf{U} = \mathbf{U}^{(t)}$ and $\mathbf{V} = \mathbf{V}^{(t)}$.

4.3. Convergence Analysis

Lemma 1. *$F(\mathbf{U})$ and $F(\mathbf{V})$ monotonically decrease when updating $\mathbf{U}$ and $\mathbf{V}$ according to Eq.* (8) *and Eq.* (13)*, respectively.*

Lemma (1) can be proved in a similar way as in [17]. This lemma assures the convergence of the problems in Eq. (5) and Eq. (10), which are the ℓ_1-norm reweighted ℓ_2-norm version of the two subproblems of RNGE in Eq. (4). We next prove the convergence of Algorithm 1 in solving the primary ℓ_1-norm objective function in Eq. (4).

Theorem 1. *The Algorithm 1 monotonically decreases the objective function of RNGE in Eq.* (4) *in each iteration and converges to an optimal solution.*

Proof. Since the objective function of RNGE in Eq. (4) is bounded below, we only need to prove the objective function monotonically decreases under the updates of $\mathbf{U}^{(t+1)}$ and $\mathbf{V}^{(t+1)}$ in each alternation. We first prove the objective function monotonically decreases under the update of $\mathbf{U}^{(t+1)}$ in Eq. (8).

According to Lemma (1), we have $F(\mathbf{U}^{(t+1)}) \leq F(\mathbf{U}^{(t)})$, i.e.,

$$\begin{aligned}
&\sum_{i,j}\left\|\mathbf{Q}^{1(t+1)}\mathbf{V}^1_{\cdot i}-\mathbf{Q}^{1(t+1)}\mathbf{V}^1_{\cdot j}\right\|_2^2\frac{\mathbf{S}_{ij}}{\left\|\mathbf{Q}^{1(t+1)}\mathbf{V}^1_{\cdot i}-\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot j}\right\|_2}+\\
&\sum_{i,j}\left\|\mathbf{Q}^{2(t+1)}\mathbf{V}^2_{\cdot i}-\mathbf{Q}^{2(t+1)}\mathbf{V}^2_{\cdot j}\right\|_2^2\frac{\mathbf{S}^p_{ij}}{\left\|\mathbf{Q}^{2(t+1)}\mathbf{V}^2_{\cdot i}-\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot j}\right\|_2}+\\
&\sum_{i,j}\frac{\left|\mathbf{X}_{ij}-(\mathbf{U}^{(t+1)}\mathbf{V})_{ij}\right|^2\lambda}{\left|\mathbf{X}_{ij}-(\mathbf{U}^{(t)}\mathbf{V})_{ij}\right|}\leq\\
&\sum_{i,j}\left\|\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot i}-\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot j}\right\|_2^2\frac{\mathbf{S}_{ij}}{\left\|\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot i}-\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot j}\right\|_2}+\\
&\sum_{i,j}\left\|\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot i}-\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot j}\right\|_2^2\frac{\mathbf{S}^p_{ij}}{\left\|\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot i}-\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot j}\right\|_2}+\\
&\sum_{i,j}\left|\mathbf{X}_{ij}-(\mathbf{U}^{(t)}\mathbf{V})_{ij}\right|^2\frac{\lambda}{\left|\mathbf{X}_{ij}-(\mathbf{U}^{(t)}\mathbf{V})_{ij}\right|}
\end{aligned}\tag{14}$$

For the sake of simplicity, we define $\mathbf{O}^{1(t)}=\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot i}-\mathbf{Q}^{1(t)}\mathbf{V}^1_{\cdot j}$, $\mathbf{O}^{2(t)}=\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot i}-\mathbf{Q}^{2(t)}\mathbf{V}^2_{\cdot j}$ and $\mathbf{Z}^{(t)}=\mathbf{X}_{ij}-(\mathbf{U}^{(t)}\mathbf{V})_{ij}$. By expanding $(\|\mathbf{O}^{1(t+1)}\|-\|\mathbf{O}^{1(t)}\|_2)^2\geq 0$, we have the following inequality:

$$\left\|\mathbf{O}^{1(t+1)}\right\|_2-\frac{\left\|\mathbf{O}^{1(t+1)}\right\|_2^2}{2\left\|\mathbf{O}^{1(t)}\right\|_2}\leq\left\|\mathbf{O}^{1(t)}\right\|_2-\frac{\left\|\mathbf{O}^{1(t)}\right\|_2^2}{2\left\|\mathbf{O}^{1(t)}\right\|_2}\tag{15}$$

Since $\mathbf{S}_{ij}\geq 0$, we multiply it on the both sides of Ineq. (15) and sum Ineq. (15) over i,j. Then we obtain

$$\begin{aligned}
&\sum_{i,j}\left(\left\|\mathbf{O}^{1(t+1)}\right\|_2\mathbf{S}_{ij}-\frac{\left\|\mathbf{O}^{1(t+1)}\right\|_2^2\mathbf{S}_{ij}}{2\left\|\mathbf{O}^{1(t)}\right\|_2}\right)\leq\\
&\sum_{i,j}\left(\left\|\mathbf{O}^{1(t)}\right\|_2\mathbf{S}_{ij}-\frac{\left\|\mathbf{O}^{1(t)}\right\|_2^2\mathbf{S}_{ij}}{2\left\|\mathbf{O}^{1(t)}\right\|_2}\right)
\end{aligned}\tag{16}$$

Similarly, we can obtain another two inequalities similar to Ineq. (16) for $\mathbf{O}^2$ and $\mathbf{Z}$. Adding such three inequalities and Ineq. (14) on whose both sides we multiply by two, we can have:

$$\begin{aligned}
&\sum_{i,j}\left\|\mathbf{O}^{1(t+1)}\right\|_2\mathbf{S}_{ij}+\sum_{i,j}\left\|\mathbf{O}^{2(t+1)}\right\|_2\mathbf{S}^p_{ij}+\sum_{i,j}\lambda\left|\mathbf{Z}^{(t+1)}\right|\leq\\
&\sum_{i,j}\left\|\mathbf{O}^{1(t)}\right\|_2\mathbf{S}_{ij}+\sum_{i,j}\left\|\mathbf{O}^{2(t)}\right\|_2\mathbf{S}^p_{ij}+\sum_{ij}\lambda\left|\mathbf{Z}^{(t)}\right|
\end{aligned}\tag{17}$$

The right hand side and left hand side of the above Ineq. (17) correspond to objective function of RNGE under $\mathbf{U}^{(t+1)}$ and $\mathbf{U}^{(t)}$, respectively. This implies the objective function monotonically decreases under the update of $\mathbf{U}$ for a given $\mathbf{V}$. Similarly, we can also prove the objective function monotonically decreases under the update of $\mathbf{V}$ for a given $\mathbf{U}$. As a result, we conclude that Algorithm 1 converges to an optimal solution in Eq. (4). □

5. Experiments

In this section, we systematically evaluated the effectiveness and robustness of our proposed Robust Nonnegative Graph Embedding (RNGE) framework. We compared RNGE to the popular subspace learning algorithms including Principal Component Analysis (PCA) [20], PCA based on ℓ_1-norm maximization (PCAL1) [14], Linear Discriminant Analysis (LDA) [1], and Marginal Fisher Analysis (MFA) [24], and other non-negative data factorization algorithms Non-negative Matrix Factorization (NMF) [15], Localized NMF (LNMF) [16] and Non-negative Graph Embedding (NGE) [21]. First, we compared the classification capabilities of these algorithms on four clean datasets. Second, we investigated their robustness to noisy data. Third, we reported how the three graph-based algorithms, i.e., RNGE, NGE, and MFA, perform towards unreliable graphs. Fourth, we investigated the robustness to noisy labels of the four supervised learning algorithms, i.e., RNGE, NGE, MFA, and LDA. Finally, we compared the algorithmic convergence of NGE and RNGE.

5.1. Data and Setting

We used three benchmark face datasets[1]: CMU **PIE**, **Yale-B**, **ORL** and one recently-released Pittsburgh **Food** Image Dataset [4] in our experiments.

PIE. The CMU PIE corpus contains 41,368 face images of 68 subjects under 13 different poses, 43 different illumination conditions, and with four different expressions. In our experiments, a subset of five near frontal poses (C05, C07, C09, C27, C29) and the illuminations indexed as 08 and 11 is used. The subset consists of 680 images in total with 10 images per subject.

Yale-B. This dataset contains 161,289 gray images of 38 subjects under nine poses and 64 illumination conditions. We chose the frontal pose and use all the images under different illumination. This gives rise to 2,432 images in total

[1] http://www.face-rec.org/databases/

with 64 images per subject.

ORL. This dataset contains 40 distinct subjects each with 10 images. All the images were used in our experiments.

Food. The data was collected by obtaining three instances of 101 foods from 11 popular fast food chains. Six images per instance were captured from different view points in both restaurant conditions and a controlled lab setting. We focus on the set of 61 food categories (e.g., Subway Chicken Wrapper). This gives rise to 1098 images with 18 images per category.

All the face images were normalized to 32-by-32 pixels. We pre-processed the face images using histogram equalizations and scaled every pixel intensity into [0,1]. For each face image, a column-stacked 1024-d vector containing pixel intensities was used as the visual feature. For each food image, we used a 2048-d Locality-constrained Linear Coding (LCC) feature [22].

For all the datasets, half of the images for each class were randomly selected as training data and the other half for testing. The reported accuracy was averaged over five random splits of all the data. For the purpose of evaluating the robustness of algorithms to noisy data, 50% training and testing images were randomly selected and occluded with noises consisting of random black and white dots. For face images, the size of occlusion patch was varied as $\{6 \times 6, 8 \times 8, 10 \times 10, 12 \times 12, 14 \times 14, 16 \times 16\}$ pixels. For food images, we set the size of noise area as $\{5\%, 10\%, ..., 30\%\}$ of the image size, as the food images are not squares. Furthermore, in order to simulate realistic occlusions, four real objects were used as occlusions in the size of 16×16 pixels for face images and 30% image size for the food images. They are piggy bank (**pig**), strawberry (**berry**), cup (**cup**) and coke can (**coke**). Fig.2 shows some typical examples of the dataset with and without noises.

For PCA, PCAL1 and MFA, the dimensions of the subspaces were tuned within $r \in \{1, 2, \cdots, m\}$, where m is the original feature dimension. For the sake of efficient evaluation, the dimensions for NMF, LNMF, NGE and RNGE were tuned within $r \in \{6\times6, 7\times7, ..., 12\times12\}$ [17]. Particularly, for NGE and RNGE, the dimension of the "intrinsic" subspace was set as $d = 0.6 \times r$, while that of the "penalty" space was $r - d$. We reported the best results by exploring all the candidate feature dimensions for all algorithms [24]. The trade-off parameter λ in NGE and RNGE was tuned empirically among $\{0.2, 0.4, 0.6, ..., 2.4\}$. For LDA and MFA, we reduced the data to the dimensions of $m - N_c$ [24], where N_c is the class number. To expedite PCAL1, PCA was performed to exclude the null space [14]. For non-negative algorithms, the reconstruction coefficients for a testing sample were computed as in [16]. For graph-based algorithms, we adopted the binary edge weights. For classification, the nearest neighbor classifier was adopted.

Figure 2. Sample images from PIE (1st row), Yale-B (2nd row), ORL (3rd row) and Food (4th row). The clean images are shown in the first column. The corrupted images with black-white occlusions are illustrated from the second to the last column with varying sizes of occlusion patches. The realistic occlusions made by pig, berry, cup and coke are shown at the last four columns.

5.2. Classification Capability

In this subsection, we evaluated the classification effectiveness of all the eight approaches on the four clean datasets. For graph-based algorithms, the numbers of nearest neighbors for the intrinsic and penalty graphs were fixed as $K_i = 3$ and $K_p = 20$, respectively [24]. Table. 1 shows the average calcification accuracies over all the classes. The corresponding subspace dimensions are provided in parenthesis. From the results, we can see that a) RNGE outperforms all the other approaches and shows the best classification capability; b) by using binary edge weights, MFA performs poorly; and c) PCAL1 and PCA perform nearly the same, since there is no noise and outlier in the clean datasets.

Table 1. Classification accuracies (%) of all the eight approaches on the four *clean* datasets. The values in parentheses are the dimensions that achieve the best results.

Algorithms	PIE	Yale-B	ORL	Food
PCA	58.99(269)	88.27(764)	85.60(80)	46.01(41)
PCAL1	59.94(308)	88.40(757)	85.60(194)	45.50(547)
NMF	57.37(144)	96.09(121)	83.40(36)	37.70(49)
LNMF	56.00(121)	87.28(144)	85.70(100)	48.49(49)
LDA	79.34(67)	94.74(37)	94.00(39)	40.18(60)
MFA	47.34(268)	90.40(634)	40.10(160)	28.60(41)
NGE	82.57(100)	99.47(144)	92.30(121)	44.70(36)
RNGE	**87.46(121)**	**99.76(144)**	**97.00(36)**	**50.09(36)**

5.3. Robustness to Noisy Data

Figure 3 illustrates the classification results of all the algorithms on the noisy datasets. From these results, we can obtain the following observations: a) RNGE outperforms NGE on all the noisy datasets with all kinds of occlusions. This demonstrates that RNGE is more robust to noisy data than NGE; b) compared to the other approaches, RNGE also shows better classification capability for noisy data; (c) on the three face datasets, the performance decreases with the increase of occlusion size. However, the performance degradation is not significant on the Food dataset. The rea-

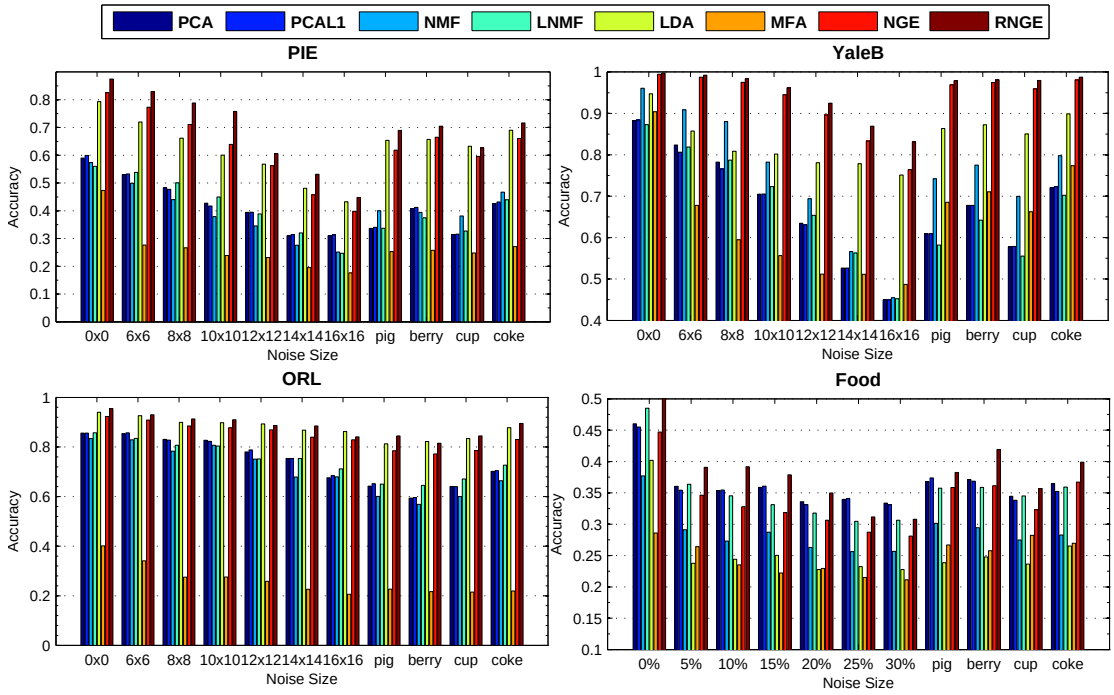


Figure 3. Classification accuracies with various sizes of occlusion patches on the four datasets. The realistic occlusions *pig, berry, cup, coke* are at the size of 16×16 pixels for face images and of 30% image area for food images. [Best Viewed in Color].

son might be that the LLC feature used for food images can suppress noises to some extend; and d) the realistic occlusions result in performance degradation compared to the results on clean images.

5.4. Robustness to Unreliable Graphs

To evaluate the robustness to unreliable graphs, we compared the three graph-based approaches, i.e., MFA, NGE, and RNGE, on the graphs constructed using various neighbor numbers. The graph becomes unreliable when the neighbor number is inappropriate for certain samples. We conducted this evaluation on Yale-B dataset, since it contains 64 images per class and thus provides sufficient scope for varying the neighbor number, while only less than 10 training or testing images per class are provided by the other three datasets. Since the nearest-neighbor parameter K_p for the penalty graph often does not affect the embedding results [24], we fixed K_p as 20, and varied the parameter K_i for the intrinsic graph from 2 to 28. Fig. 4 shows the average classification accuracies with various K_i. As we can see, while NGE and MFA change drastically, our RNGE is very stable to various K_i and achieves consistent performance improvements over NGE and MFA. Such superiority in performance demonstrates the better robustness of RNGE to unreliable graphs.

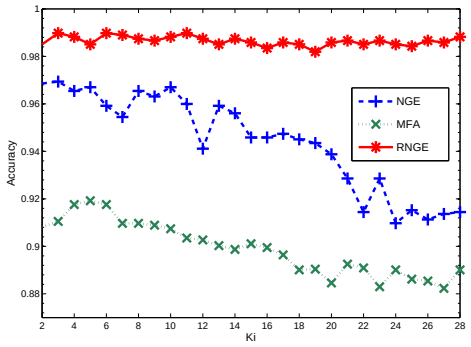


Figure 4. Classification accuracies with various neighbor numbers K_i on the YaleB. As we can see, RNGE is robust to unreliable graphs.

5.5. Robustness to Noisy Labels

In this experiment, we investigated how well the four supervised methods LDA, MFA, NGE, and RNGE perform with the presence of noisy labels in the training data. $k\%$ training samples in Yale-B dataset were manually mislabeled as incorrect classes. k was varied within $\{0, 10, ..., 60\}$. Fig. 5 illustrates the average classification accuracies from the four approaches. While LDA, NGE, and MFA suffer from significant performance degradation with the increase of noisy labels, RNGE shows better robustness to the noisy labels and the performance degradation is slight.

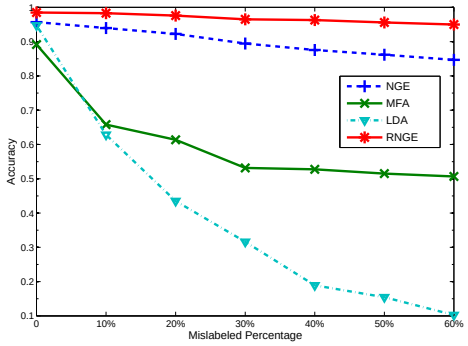
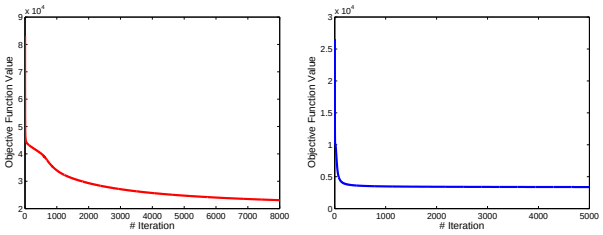


Figure 5. Classification accuracies with various amount of noisy labels on the YaleB.As we can see, RNGE is robust to noisy labels.



(a) RNGE (b) NGE

Figure 6. Convergence curves of (a) RNGE and (b) NGE on the ORL dataset.

5.6. Algorithmic Convergence

As proved in Section 4.3, the updating rules in Algorithm 1 guarantees an optimal solution for the objective function of the proposed RNGE in Eq. (4). Fig. 6(a) shows how the object function value decreases with the increase of iteration number on the ORL dataset. Compared to the convergence of NGE in Fig. 6(b), RNGE has a slightly slower convergence caused by the iterative solution of the ℓ_1-norm reweighted ℓ_2-norm problem in RNGE. For NGE and RNGE, we here set the same convergence criteria, i.e., both the norms of $\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}$ and $\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}$ are less than $\sqrt{m \times n} \times 10^{-5}$. In the experiments on the four datasets, NGE converges in 3000-5000 iterations, while and RNGE converges in 4000-8000 iterations.

6. Conclusions

We have proposed a novel Robust Non-negative Graph Embedding (RNGE) framework, which possesses robustness to noisy data, unreliable graphs, and noisy labels. Different from the traditional NGE based on ℓ_2-norm, RNGE formulates the graph embedding and data reconstruction functions using ℓ_1-norm. While the sparse solution of ℓ_1-norm reconstruction function alleviates the influence of noisy data, the ℓ_1-norm embedding functions lead to sparse and more ideal embedding results, and hence endow the robustness of RNGE to unreliable graphs and noisy labels. An efficient multiplicative updating algorithm has been develop to solve the ℓ_1-norm minimization and the convergence is guaranteed with theoretical analysis. Extensive experiments compared with the representative non-negative data factorization and subspace learning approaches on four real-world datasets have demonstrated the effectiveness and robustness of the proposed RNGE.

Acknowledgments

This work was supported by NUS-Tsinghua Extreme Search (NExT) project under the grant number: R-252-300-001-490.

References

[1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *TPAMI*, 1997.

[2] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. ICDM, 2008.

[3] E. L. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *JACM*, 2011.

[4] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. ICIP, 2009.

[5] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: rotational invariant l1-norm principal component analysis for robust subspace factorization. ICML, 2006.

[6] D. Field. What is the goal of sensory coding? *Neural computation*, 1994.

[7] Q. Gu and J. Zhou. Local learning regularized nonnegative matrix factorization. IJCAI, 2009.

[8] T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3d non-negative tensor factorization. ICCV, 2005.

[9] X. He and P. Niyogi. Locality preserving projections. NIPS, 2004.

[10] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 2004.

[11] Y. Jia and T. Darrell. Heavy-tailed distances for gradient based image descriptors. NIPS, 2011.

[12] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. CVPR, 2005.

[13] D. Kong, C. Ding, and H. Huang. Robust nonnegative matrix factorization using l21-norm. CIKM, 2011.

[14] N. Kwak. Principal component analysis based on l1-norm maximization. *TPAMI*, 2008.

[15] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. NIPS, 2001.

[16] S. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng. Learning spatially localized, parts-based representation. CVPR, 2001.

[17] X. Liu, S. Yan, and H. Jin. Projective nonnegative graph embedding. *TIP*, 2010.

[18] F. Nie, W. Hua, H. Huang, and C. Ding. Unsupervised and semi-supervised learning via l1-norm graph. ICCV, 2011.

[19] Y. Pang and Y. Yuan. Outlier resisting graph embedding. *Neurocomputing*, 2010.

[20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 1991.

[21] C. Wang, Z. Song, S. Yan, L. Zhang, and H.-J. Zhang. Multiplicative nonnegative graph embedding. CVPR, 2009.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. CVPR, 2010.

[23] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *PIEEE*, 2010.

[24] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *TPAMI*, 2007.

[25] J. Yang, S. Yan, Y. Fu, X. Li, and T. Huang. Non-negative graph embedding. CVPR, 2008.