remote sensing

MDPI

Article

Fast Spectral Clustering for Unsupervised Hyperspectral Image Classification

Yang Zhao [1,2], Yuan Yuan [3,*] and Qi Wang [3]

[1] Key Laboratory of Spectral Imaging Technology CAS, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China; zhaoyang.opt@gmail.com
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China; crabwq@gmail.com
* Correspondence: y.yuan1.ieee@gmail.com

Received: 16 January 2019; Accepted: 12 February 2019; Published: date

Abstract: Hyperspectral image classification is a challenging and significant domain in the field of remote sensing with numerous applications in agriculture, environmental science, mineralogy, and surveillance. In the past years, a growing number of advanced hyperspectral remote sensing image classification techniques based on manifold learning, sparse representation and deep learning have been proposed and reported a good performance in accuracy and efficiency on state-of-the-art public datasets. However, most existing methods still face challenges in dealing with large-scale hyperspectral image datasets due to their high computational complexity. In this work, we propose an improved spectral clustering method for large-scale hyperspectral image classification without any prior information. The proposed algorithm introduces two efficient approximation techniques based on Nyström extension and anchor-based graph to construct the affinity matrix. We also propose an effective solution to solve the eigenvalue decomposition problem by multiplicative update optimization. Experiments on both the synthetic datasets and the hyperspectral image datasets were conducted to demonstrate the efficiency and effectiveness of the proposed algorithm.

Keywords: spectral clustering; hyperspectral image classification; remote sensing; manifold learning; unsupervised learning

1. Introduction

Hyperspectral images (HSIs) contain information on hundreds of continuous narrow spectral wavelengths, which are collected by aircrafts, satellites, and unmanned aerial vehicles in each HSI pixel [7–10]. Since HSIs reflect rich spectral and spatial resolution, they offer the potential to discriminate more detailed classes and provide even broader applications for land-over classification and clustering [2–5]. To a certain extent, dealing with HSIs is difficult because the numerous spectral bands significantly increase the computational complexity and the noise in HSIs can badly influence the classification accuracy [1,6]. The existing work reported by most scholars can be roughly divided into two categories according to whether a certain number of training samples are required, as demonstrated in [11,12]: (1) supervised learning named HSI classification; and (2) unsupervised learning named HSI clustering. In the literature, many HSI classification algorithms have been proposed and they have achieved excellent performances. One popular method for HSI classification is to first use dimension reduction and then follow a classifier such as support vector machines [36,37]. Due to the noises and redundancy among spectral bands, many feature extraction, band selection and dimension reduction techniques have been developed in the past years. Some representative work, such as principle component analysis [38] and feature-selection algorithm [39,50], are also widely applied in

HSI classification. Kernel-based algorithms such as SVM and its variants [37] have been shown to improve performance [44]. Sparse representation [45] has also been introduced to the task of HSI classification. Newly raised deep learning techniques [46] have proved to be useful for supervised HSI classification.

HSI classification based on supervised methods provides excellent performance on standard datasets (e.g., more than 95% of the overall accuracy) [13]. However, the reported HSI classification algorithms require a certain number of high quality samples to obtain an optimal model. Recently, many researchers noticed that it is expensive or even impossible to collect enough labeled training data in some cases, and some recent work pay more attention to the problem of "small sample size" and present encouraging results, e.g., semi-supervised learning [15], active learning [16], domain adaptation [17], and tensor learning [18]. Although these methods could achieve similar classification results as supervised ones while using fewer training samples, they are still supervised methods that require high quality training samples to learn the classification model. On the contrary, clustering-based techniques require little prior knowledge and can be considered as data preprocessing methods to provide necessary reference information regarding supervised classification, target detection, or spectral unmixing. Therefore, unsupervised HSI classification is an extremely important techniques and has attracted significant attention in recent years. Wang et al. [19] illustrated that the existing algorithms can be coarsely divided into the following four categories: (1) Centroid-based clustering methods, such as k-mean [20] and fuzzy c-means [21], minimize the within cluster sample distance, but are sensitive to initialization and noise, and cannot provide a robust performance. (2) Density-based methods include the clustering by fast search and find the density peak algorithm [22], the density-based spatial clustering of applications with noise [23], and the clustering-in-quest method [24], which are not suitable for HSIs as it is difficult to get the density peak in the sparse feature space. (3) Biological clustering methods include the artificial immune networks for unsupervised remote sensing image classification [25] and the automatic fuzzy clustering method based on adaptive multiobjective differential evolution [26]. Their results are not always satisfactory because biological models do not always exactly fit the characteristics of HSIs. (4) Graph-based methods, such as spectral clustering [14,27], perform well in the task of unsupervised HSI classification but most of them take too much time on the eigenvalue decomposition and the affinity matrix.

In general, the accuracy of the existing unsupervised HSI classification algorithms are far from satisfactory compared to the supervised techniques due to the uniform data distribution caused by the large spectral variability. In this paper, we focus on the family of graph-based clustering algorithms (i.e., spectral clustering algorithms) [32,33]. Compared with other clustering techniques, spectral clustering has good performance in dealing with irregularly-shaped clusters and gradual variation within groups. In general, spectral clustering performs a low-dimension embedding of the affinity matrix followed by a k-means clustering in the low-dimensional space [51]. The utilization of graph model and manifold information makes it possible to process the data with complicated structure. Accordingly, algorithms based on spectral clustering have been widely applied and shown their effectiveness in the task of HSI processing. Although the spectral clustering methods have performed well, it would be too expensive to calculate the pairwise distance of enormous samples and difficult to provide an optimal approximation for eigenvalue decomposition in dealing with a large affinity matrix. In the clustering process, the complexity mainly arises from two aspects. First, the storage complexity of the affinity matrix is $O(n^2)$ and the corresponding time complexity is $O(n^2d)$. The second is the eigenvalue decomposition of Laplacian matrix, which is $O(n^2c)$ time complexity. Note that n, d, and c are the number of pixels, feature dimensions, and classes of HSI, respectively. It is obvious that high spatial resolution (i.e., number of pixels n) is a major constraint to apply spectral clustering to real-life HSI applications. In our experiments, spectral clustering techniques can be applied to small-scale HSI datasets such as Samson, Jasper, SalinasA, and India Pines, as these datasets contain only about 10,000 pixels. However, along with the increase of spatial resolution of HSIs, it could be unacceptable for the large-scale HSI datasets including Salinas, Pavia University, Kennedy Space Center, and Urban,

which contain about 100,000 pixels, because of the rapid growth of the storage and time complexity of affinity matrix construction and eigenvalue decomposition of Laplacian matrix.

To alleviate the above problem, several improved spectral clustering methods have been proposed for large-scale HSIs with high spatial resolution. An efficient way to get low-rank matrix approximation based on Nyström extension has been widely applied in many kernel based clustering task [28,29], and recent studies have shown good performance in the task of HSI processing [48,49]. Another method proposed by Nie et al. [30,31] constructs anchor-based affinity matrix with balanced k-means based hierarchical k-means algorithm. Wang et al. [19] improved the anchor-based affinity matrix by incorporating the spatial information. Meanwhile, Nonnegative Matrix Factorization (NMF) technique [34,35] and its variants also provide an efficient solution for HSI classification. Motivated by the existing approaches, we propose an improved spectral clustering based on multiplicative update algorithm and two efficient methods for affinity matrix approximation. In general, the spectral clustering problem can be solved by the standard trace minimization of the objective function and we propose an efficient resolution though multiplicative update optimization according to the derivative of the objective function. Meanwhile, the nonnegative constraint and the orthonormal constraint provide a better indicator matrix and this makes it easier to get a robust clustering result by the later processing such as k-means. Furthermore, the anchor-based graph and the Nyström extension are introduced to improve the computational complexity by affinity matrix approximation for the large-scale HSIs. There are three main contributions of this work:

1. An novel multiplicative update optimization for eigenvalue decomposition is proposed for large-scale unsupervised HSIs classification. It is worth noting that the proposed method can be easily portable to the variants of spectral clustering methods with different regularization items only if the constraints are convex functions.
2. Two affinity matrix approximation techniques, namely the anchor-based graph and the Nyström extension, are introduced to improve the affinity matrix by sampling limited samples (i.e., pixels or anchors).
3. Comprehensive experiments on the HSI datasets illustrated that the proposed method achieved a good result in terms of efficiency and effectiveness, and the combination of multiplicative update method and affinity matrix approximation provided a better performance.

The rest of this paper is organized as follows. Section 2 provides notations and a brief view of the general spectral clustering algorithm. Next, we present the motivation and formulate the proposed multiplicative update algorithm. Furthermore, an effective multiplicative update method for eigenvalue decomposition is presented in Section 3. To further improve the computational complexity of affinity matrix, we introduce two efficient approximated techniques in Section 4. The experimental results including performance analyses, computational complicity and parameter determination are given in Section 5. Section 6 concludes this paper.

2. Overview

We begin by reviewing the classical spectral clustering algorithm, and before going into the details, we firstly introduce the notation.

2.1. Notation

In this part, we define some notation to make sure that the mathematical meaning of the proposed method can be formulated clearly. The pixels of HSIs can be considered as $\{\mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, ..., n\}$ where d is the dimensionality (i.e., the number of spectral bands). Let $\{\mathbf{y}_1, \mathbf{y}_1, ..., \mathbf{y}_c\} \subset \mathbb{R}^c$ be the indicator vectors of the pixels $\{\mathbf{x}_1, \mathbf{x}_1, ..., \mathbf{x}_n\}$, respectively. Here, $\mathbf{y}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, ..., \mathbf{y}_{ic}]$, where c is the predefined number of clusters, and the indicator vectors $\mathbf{y}_{ij} = 1$ if and only if $\mathbf{x}_i$ belongs to the jth cluster and $\mathbf{y}_{ij} = 0$ otherwise. Denote $\mathbf{Y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_n^T]^T \in \mathbb{R}^{n \times c}$, and $\mathbf{Y} \geq 0$ indicates that the whole elements of $\mathbf{Y}$ are nonnegative. The affinity matrix is denoted by $\mathbf{W}$ and the Laplacian matrix

is denoted by $\mathbf{L}$. The corresponding trace can be denoted by $\mathrm{Tr}(\mathbf{W})$ and the Frobenius norm of $\mathbf{W}$ is denoted by $||\mathbf{W}||_F$. The detailed notations are summarized in Table 1 and we explain the meaning of each term when it is first used.

Table 1. Notation.

$\mathbf{W}$	Affinity (or similarity) matrix
$\mathbf{D}$	Diagonal matrix
$\mathbf{L}$	Laplacian matrix
$\mathbf{Y}$	Cluster indicator matrix
$\mathbf{y}$	Cluster indicator
$\mathbf{x}$	Pixels (or data points)
$\mathbf{I}$	Identity matrix
n	Number of pixels
m	Number of chosen pixels (or anchors)
d	Number of spectral bands
c	Number of classes

2.2. Normalized Cuts Revisit

A set of samples (i.e., pixels) $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ can be considered as an undirected graph $G = \{Vertices, Edges\}$. Each vertex represents a sample $\mathbf{x}_i$ and the edge is aligned by their similarity. In general, the corresponding affinity (or similarity) matrix $\mathbf{W}$ can be denoted as

$$\mathbf{W}_{ij} = e^{\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}}, \qquad i, j = 1, 2, ..., n, \tag{1}$$

where σ is the width of the neighbors, $\mathbf{W}$ is a symmetric matrix and $\mathbf{W}_{ij}$ is the affinity of samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Let A and B represent a bipartition of $Vertices$, where $A \cup B = Vertices$ and $A \cap B = \emptyset$. Let $\mathrm{cut}(A, B)$ denotes the sum of the weights between A and B as $\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} \mathbf{W_{ij}}$. The volume of a set is defined as the sum of the degrees within that set: $\mathrm{vol}(A) = \sum_{i \in A} \mathbf{D}_{ii}$ and $\mathrm{vol}(B) = \sum_{i \in B} \mathbf{D}_{ii}$, where $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. The normalized cut between A and B can be considered as follows:

$$\mathrm{NCut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(A)} + \frac{\mathrm{cut}(A, B)}{\mathrm{vol}(B)} = \frac{2\mathrm{cut}(A, B)}{\mathrm{vol}(A)||\mathrm{vol}(B)}, \tag{2}$$

where $||$ is the harmonic mean. According to Author1 [41], an optimal resolution of $\mathrm{NCut}(A, B)$ can be provided by solving the minimization of the following equation

$$\min \frac{\mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} = \min \mathbf{y}^T\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{y}, \tag{3}$$

where $\mathbf{D}$ is the diagonal matrix with elements $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. $\mathbf{y}$ is the indicator vector, where $\mathbf{y}_{ij} = 1$ if and only if $\mathbf{x}_i$ belongs to the jth cluster and $\mathbf{y}_{ij} = 0$ otherwise.

According to spectral graph theory, an approximate resolution of Equation (3) can be considered as thresholding the eigenvector corresponding to the second smallest eigenvalues of the normalized Laplacian $\mathbf{L}$ as follows:

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}. \tag{4}$$

Shi and Malik [41] illustrated that the normalized Laplacian matrix $\mathbf{L}$ is positive semidefinite even when $\mathbf{W}$ is indefinite. Its second smallest eigenvalue lies on the interval $[0, 2]$ so the corresponding eigenvalues of $\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ are confined to lie inside $[-1, 1]$. Considering the case of multiple group clustering where $c > 2$, Equation (3) can be rewritten as

$$\min \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}), \tag{5}$$

where $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$ and $\mathbf{Y}$ is the indicator matrix. This can be solved by the standard trace minimization problem according to the normalized spectral clustering proposed in [41]. The solution $\mathbf{Y}$ consists of the top c eigenvectors of the normalized Laplacian matrix $\mathbf{L}$ as columns.

However, there are two tough problems to get an efficient and effective solution by using the classical spectral clustering technique: one is the eigenvalue decomposition of the Laplacian matrix $\mathbf{L}$, which takes $O(n^2c)$ time complexity, and the other one is the storage and time complexity of the affinity matrix, which are $O(n^2)$ and $O(n^2d)$, respectively. It is obvious that either of the above problems can be an unbearable burden with the increasing of the number of samples. To alleviate the above problem, motivated by the recent work such as Nyström extension, anchor-based graph and nonnegative matrix factorization, we propose a novel approach to solving the large-scale and high-dimensional HSI clustering (or unsupervised HSI classification), and the detailed demonstration are presented in the following sections.

3. Improved Spectral Clustering with Multiplicative Update Algorithm

In this section, we propose an multiplicative update algorithm to get an efficient resolution of the eigenvalue decomposition of the Laplacian matrix $\mathbf{L}$. We firstly present the formulation and our motivation, and then a novel resolution for spectral clustering based on multiplicative update algorithm is proposed in Section 3.2.

3.1. Formulation and Motivation

In general, a multigroup spectral clustering problem (i.e., $c > 2$) can be considered as a minimization of the following equation:

$$\min \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) + \lambda||\mathbf{Y}^T\mathbf{Y} - \mathbf{I}||_F^2, \tag{6}$$

where $\lambda > 0$ is the Lagrangian multiplier and $||\mathbf{Y}^T\mathbf{Y} - \mathbf{I}||_F^2$ is the item for orthonormal constraint. However, Equation (6) is still a non-smooth objective function, thus it is difficult to obtain an efficient resolution by solving the eigenvalue decomposition of the Laplacian matrix $\mathbf{L}$. Motivated by NMF, which has excellent performance in dealing with clustering by relaxation technique, we relax the discreteness condition and propose an multiplicative update optimization to solve the eigenvalue decomposition, the details of which are illustrated in the next section.

3.2. Multiplicative Update Optimization

Spectral clustering cannot provide an efficient resolution since it would be too expensive to get an optimal approximation for eigenvalue decomposition in deal with large-scale datasets. Motivated by the recent work on NMF, we introduce the nonnegative constraint for indicator matrix as $\mathbf{Y}$ where $\mathbf{Y}_{ij} > 0$. Moreover, the traditional spectral relation approaches relax the indicator matrix $\mathbf{Y}$ to orthonormal constraint as $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$. According to a recent work [47], if the indicator matrix $\mathbf{Y}$ is orthonormal and nonnegative simultaneously, only one element is positive and the others are zeros in each row of $\mathbf{Y}$. Note that we can get an ideal indicator matrix $\mathbf{Y}$ as defined in Section 2.1 by considering the above two constraints: $\mathbf{Y} > 0$ and $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$. The above constraints are significant, which can help us to solve the eigenvalue decomposition in a more efficient way and this is also easy to implement.

By relaxing the discreteness condition and considering the above two constraints, Equation (6) can be rewritten as

$$\min \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) + \lambda(\mathbf{Y}^T\mathbf{Y} - \mathbf{I}) = \min \mathrm{Tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) + \lambda \mathrm{Tr}((\mathbf{Y}^T\mathbf{Y} - \mathbf{I})^T(\mathbf{Y}^T\mathbf{Y} - \mathbf{I})), \tag{7}$$

where $\mathbf{Y} > 0$. Equation (7) can be considered as the cost function and we try to find an optimal resolution of minimization. The derivation of Equation (7) is

$$\mathbf{L}\mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y} - 2\lambda\mathbf{Y}, \tag{8}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ and Equation (8) can be rewritten as

$$\begin{aligned}&(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})\mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y} - 2\lambda\mathbf{Y}\\ =&\mathbf{Y} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y} - 2\lambda\mathbf{Y}\\ =&(\mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y}) - (2\lambda\mathbf{Y} + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\mathbf{Y}).\end{aligned} \tag{9}$$

In this case, the derivation of Equation (7) is divided into two parts. Both $\mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y}$ and $2\lambda\mathbf{Y} + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\mathbf{Y}$ are nonnegative matrices since $\mathbf{Y} > 0$, $\mathbf{D} > 0$, and $\mathbf{W} \geq 0$. For convenience, we denote the former factor as $\mathbf{Q} = \mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y}$ and the latter factor as $\mathbf{P} = 2\lambda\mathbf{Y} + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\mathbf{Y}$. According the multiplication update rule for standard NMF algorithm [42], we can get the minimization of the cost function in Equation (7) by updating $\mathbf{Y}$ as follows:

$$\mathbf{Y} \leftarrow \mathbf{Y} \circ \mathbf{P} \oslash \mathbf{Q}, \tag{10}$$

where $\circ$ and $\oslash$ denote Hadamard product and Hadamard division (i.e., element-wise multiplication and division), respectively, and $\mathbf{Y}_{ij} \leftarrow \mathbf{Y}_{ij} \cdot \mathbf{P}_{ij}/\mathbf{Q}_{ij}$. Then, we can get a optimal resolution until the cost function converge and the implement details are presented in Algorithm 1. Since only one element is positive and the others approximate zero in each row of the indicator matrix $\mathbf{Y}$, it can be considered as a nearly perfect indicator matrix for clustering representation.

Algorithm 1: Algorithm to solve the problem in Equation (6).

Input: Hyperspectral image datasets $\mathbf{X}$.
Output: Indicator matrices $\mathbf{Y}$ and clustering result $\mathbf{S}$.
Initialize indicator matrix randomly such that $\mathbf{Y} > 0$.
Choose m samples in $\mathbf{X}$:
(a). If using Nyström extension, calculate the matrices $\mathbf{A}$ and $\mathbf{B}$ by Equation (17).
(b). If using anchor-based graph, calculate the matrix $\mathbf{Z}$ according to Equation (26).
while *Equation (6) not converge* **do**

1. Update numerator matrix $\mathbf{P}$ and denominator matrix $\mathbf{Q}$:
   (a). If using Nyström extension, update $\mathbf{P}$ and $\mathbf{Q}$ with $\mathbf{A}$ and $\mathbf{B}$ by Equation (19).
   (b). If using anchor-based graph, update $\mathbf{P}$ and $\mathbf{Q}$ with $\mathbf{Z}$ by Equation (28).
2. Update the indicator matrix $\mathbf{Y}$ according to Equation (10):
   $\mathbf{Y}_{ij} = \mathbf{Y}_{ij}\sqrt{\frac{\mathbf{P}_{ij}}{\mathbf{Q}_{ij}}}$.

end
Input $\mathbf{Y}$ to k-means to get the clustering result $\mathbf{S}$.

4. Approximated Affinity Matrix

To further improve the time and storage complexity of computing affinity matrix to make the spectral clustering algorithm available for large-scale datasets such as HSIs, we introduce anchor-based graph and Nyström extension to approximate the original affinity matrix with limited samples.

4.1. Affinity Matrix with Nyström Extension

The Nyström extension is a technique for finding numerical approximations to eigenfunction problems and the detailed illustration can be found in [40]. It allows us to extend an eigenvector computed for a set of sample points to arbitrary samples $\mathbf{x}$ with the interpolation weights.

Inspired by Author1 [41], the affinity matrix considers both the brightness value of the pixels and their spatial location, and we can define the similarity of two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$\mathbf{W}_{ij} = e^{\frac{-||\mathbf{l}_i - \mathbf{l}_j||_2^2}{2\sigma_l^2}} \cdot e^{\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma_x^2}}, \tag{11}$$

where $\mathbf{l}_i$ and $\mathbf{l}_j$ are the spatial locations of the HSI's pixels. σ_l and σ_x are the bandwidth of neighboring pixels and these parameters are sensitive to different HSIs. To alleviate this problem, Zhao et al. [27] introduced an adaptive parameter and we can define $\bar{\sigma}_l$ and $\bar{\sigma}_x$ as

$$\begin{aligned} \bar{\sigma}_l^2 &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} ||\mathbf{l}_i - \mathbf{l}_j||_2^2, \\ \bar{\sigma}_x^2 &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} ||\mathbf{x}_i - \mathbf{x}_j||_2^2. \end{aligned} \tag{12}$$

and Equation (11) can be presented as

$$\mathbf{W}_{ij} = e^{\frac{-||\mathbf{l}_i - \mathbf{l}_j||_2^2}{2\alpha\bar{\sigma}_l^2}} \cdot e^{\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\alpha\bar{\sigma}_x^2}}. \tag{13}$$

where the parameter α controls the neighbor of affinity matrix.

For uniformity in notation, the affinity matrix $\mathbf{A}$ is similarity defined by Equation (11) of m chosen samples. The affinity matrix of the remaining $n - m$ samples and the chosen samples are denoted as $\mathbf{B}$. $\mathbf{C}$ is the affinity matrix for the remaining samples. The affinity matrix $\mathbf{W}$ can be rewritten as

$$\mathbf{W} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}, \tag{14}$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B} \in \mathbb{R}^{m \times (n-m)}$ and $\mathbf{C} \in \mathbb{R}^{(n-m) \times (n-m)}$. According to the Nyström extension, $\mathbf{C}$ can be denoted by $\mathbf{C} = \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ and the approximated affinity matrix $\mathbf{W}$ can be formulated as

$$\widehat{\mathbf{W}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}. \tag{15}$$

We can find that the size of this norm is governed by the extent to which $\mathbf{C}$ is spanned by the rows of $\mathbf{B}$, and the Nyström extension provides an approximation to the entire affinity matrix with only a subset of rows or columns.

To extend the above matrix form of Nyström method to NCut, we need to calculate the row sum of matrix $\widehat{\mathbf{W}}$. However, it is possible without explicitly evaluating the sub-matrix $\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ since

$$\mathbf{d} = \widehat{\mathbf{W}}\mathbf{1} = \begin{bmatrix} \mathbf{A}\mathbf{1}_m + \mathbf{B}\mathbf{1}_n \\ \mathbf{B}^T\mathbf{1}_m + \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}\mathbf{1}_n \end{bmatrix}, \tag{16}$$

where $\mathbf{A}\mathbf{1}_m$ and $\mathbf{B}\mathbf{1}_n$ are the row sum of matrix $\mathbf{A}$ and $\mathbf{A}$ and $\mathbf{B}^T\mathbf{1}_m$ is the column sum of matrix $\mathbf{B}$. Then, the matrix $\mathbf{A}$ and $\mathbf{B}$ can be formulated as

$$\begin{aligned} \mathbf{A}_{ij} &\leftarrow \frac{\mathbf{A}_{ij}}{\sqrt{\mathbf{d}_i\mathbf{d}_j}}, \\ \mathbf{B}_{ij} &\leftarrow \frac{\mathbf{B}_{ij}}{\sqrt{\mathbf{d}_i\mathbf{d}_{j+m}}}, \end{aligned} \tag{17}$$

and we can get the normalized affinity matrix $\mathbf{D}^{-\frac{1}{2}}\widehat{\mathbf{W}}\mathbf{D}^{-\frac{1}{2}}$ (refer to Equation (15)) as before; thus, we can get

$$\mathbf{D}^{-\frac{1}{2}}\widehat{\mathbf{W}}\mathbf{D}^{-\frac{1}{2}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}, \tag{18}$$

where $\mathbf{A}$ and $\mathbf{B}$ are from Equation (17). However, the elements of $\mathbf{D}^{-\frac{1}{2}}\widehat{\mathbf{W}}\mathbf{D}^{-\frac{1}{2}}$ can be negative since the matrix $\mathbf{A}^{-1}$ may contain negative elements. However, we have to keep $\mathbf{D}^{-\frac{1}{2}}\widehat{\mathbf{W}}\mathbf{D}^{-\frac{1}{2}} \geq 0$ to satisfy the constraints of the proposed multiplicative update algorithm. Because of this, we denote $\mathbf{A}^+ = (|\mathbf{A}| + \mathbf{A})./2$ and $\mathbf{A}^- = (|\mathbf{A}| - \mathbf{A})./2$, where we can find that $\mathbf{A}^+$ is the positive part of $\mathbf{A}$ and $\mathbf{A}^-$ is the negative part of $\mathbf{A}$. Note that both $\mathbf{A}^+$ and $\mathbf{A}^-$ are negative matrix and $\mathbf{P}$ and $\mathbf{Q}$ can be formulated as

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^+ \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \mathbf{Y} + 2\lambda\mathbf{Y}, \\ \mathbf{Q} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^- \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} + \mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y}. \end{aligned} \tag{19}$$

4.2. Affinity Matrix with Anchor-Based Graph

The anchor-based graph was proposed by Zhu et al. [30] for large-scale data clustering problem. It makes the label prediction function a weighted average of the labels on a subset of anchor samples. If one can infer the labels of other unlabeled samples, they are easily obtained by a simple linear combination. As such, the label prediction function $f(\cdot)$ can be represented by a subset $\mathbf{A} = \{\mathbf{a}_j\}_{j=1}^m \subset \mathbb{R}^D$ in which each $\mathbf{a}_j$ acts as an anchor sample,

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} \mathbf{Z}_{ij} f(\mathbf{a}_j), \tag{20}$$

where $\mathbf{Z}$ is the data-adaptive weight matrix which measures the similarity between samples and anchors. We define two vectors $\mathbf{F} = [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_n)]^T$ and $\mathbf{F}_a = [f(\mathbf{a}_1), f(\mathbf{a}_2), ..., f(\mathbf{a}_n)]^T$, thus Equation (20) can be rewritten as

$$\mathbf{F} = \mathbf{Z}\mathbf{F}_a, \mathbf{Z} \in \mathbb{R}^{n \times m}, m \ll n. \tag{21}$$

This formula reduces the solution space of unknown labels from large $\mathbf{F}$ to smaller $\mathbf{F}_a$.

The first problem of anchor-based graph construction is how to choose the anchors. In general, the anchors can be considered as random samples or representative samples such as k-means clustering centers. Random selection chooses m anchors by random sampling from samples with computational complexity $O(1)$. However, the randomly chosen samples cannot guarantee that the approximated affinity matrix is always robust. Liu et al. [43] suggested using k-means clustering centers as anchors instead of randomly chosen samples since the k-means clustering centers have a robust representation power to adequately cover the whole data. However, the computational complexity of k-means is $O(ndmt)$, where t is the number of iterations.

The second problem is how to design a regression matrix $\mathbf{Z}$ that measure the underlying relationship between the whole samples and the chosen anchors. Liu et al. [43] proposed a method named Local Anchor Embedding (LAE) to reconstruct the regression matrix, where $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_m\}$ denote the chosen anchors and $K(\cdot)$ is a given kernel function with bandwidth parameters:

$$\mathbf{Z}_{ij} = \frac{K(\mathbf{x}_i, \mathbf{a}_j)}{\sum_{k \in \Phi_i} K(\mathbf{x}_i, \mathbf{a}_k)}, \forall j \in \Phi_i \tag{22}$$

The notation $\Phi_i \subset [1, 2, ..., m]$ is the set saving the indexes of s nearest neighbors of $\mathbf{x}_i$ in $\mathbf{A}$, and the Gaussian kernel $K(\mathbf{x}_i, \mathbf{a}_j) = \exp(-||\mathbf{x}_i - \mathbf{a}_j||_2^2/2\sigma^2)$ is adopted for the kernel regression. However, the kernel-based methods need an extra parameter (i.e., bandwidth σ). Nie et al. [20] adopted a parameter-free yet effective neighbor assignment strategy and they obtained the ith row of $\mathbf{Z}$ by solving following problem:

$$\min_{\mathbf{Z}_i^T\mathbf{1}=1, \mathbf{Z}_i \geq \mathbf{0}} \sum_{j=1}^{m} ||\mathbf{x}_i - \mathbf{a}_j||_2^2 \mathbf{Z}_{ij} + \gamma \mathbf{Z}_{ij}^2, \tag{23}$$

where $\mathbf{Z}_i^T$ denotes the ith row of $\mathbf{Z}$ and γ is the regularization parameter. Note that Equation (23) does not consider the spatial information of HSIs, which may result in some isolated pixels appearing in the clustering HSI due to the existence of noise, outliers, or mixed pixels. Recent studies incorporate the spatial information by directly modifying the cost function in Equation (23) as follows:

$$\min_{\mathbf{Z}_i^T\mathbf{1}=1, \mathbf{Z}_i \geq \mathbf{0}} \sum_{j=1}^{m} ||\mathbf{x}_i - \mathbf{a}_j||_2^2 \mathbf{Z}_{ij} + \beta ||\bar{\mathbf{x}}_i - \mathbf{a}_j||_2^2 \mathbf{Z}_{ij} + \gamma \mathbf{Z}_{ij}^2, \tag{24}$$

where $\bar{\mathbf{x}}_i$ is the mean of the neighboring pixels lying within a window around $\mathbf{x}_i$ and the parameter α controls the tradeoff between hyperspectral information and spatial information. Let $\mathbf{d}_{ij}^x = ||\mathbf{x}_i - \mathbf{u}_j||_2^2$ and $\mathbf{d}_{ij}^{\bar{x}} = ||\bar{\mathbf{x}}_i - \mathbf{u}_j||_2^2$, and denote $\mathbf{d}_i \in \mathbb{R}^m$ a vector with the jth element as $\mathbf{d}_{ij} = \mathbf{d}_{ij}^x + \beta \mathbf{d}_{ij}^{\bar{x}}$. It is obvious that Equation (24) can be rewritten in vector form as

$$\min_{\mathbf{Z}_i} ||\mathbf{Z}_i + \frac{1}{2\gamma}\mathbf{d}_i||_2^2, \tag{25}$$

where $\mathbf{Z}_i^T\mathbf{1} = 1$ and $\mathbf{Z}_i \geq \mathbf{0}$. Following y Nie et al. [30], the parameter γ can be denoted by $\gamma = (s/2)\mathbf{d}_{i,s+1} - (1/2)\sum_{j=1}^{s} \mathbf{d}_{ij}$, and the resolution of Equation (25) is

$$\mathbf{Z}_{ij} = \frac{\mathbf{d}_{i,k+1} - \mathbf{d}_{ij}}{k\mathbf{d}_{i,k+1} - \sum_{j'=1}^{k} \mathbf{d}_{ij'}}. \tag{26}$$

For the detailed deviation, see [20]. After we get the regression matrix $\mathbf{Z}$, the affinity matrix $\mathbf{W}$ can be obtained as

$$\widehat{\mathbf{W}} = \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{Z}^T, \tag{27}$$

where $\mathbf{\Delta}$ is a diagonal matrix, the jth entry is defined as $\sum_{i=1}^{n} \mathbf{Z}_{ij}$ and $\widehat{\mathbf{W}}$ is symmetric positive semidefinite and doubly stochastic. Not that $\mathbf{Z}_i^T\mathbf{1} = 1$ and $\mathbf{Z}_i \geq \mathbf{0}$, and it can be verified that $\widehat{\mathbf{W}}$ is a double stochastic matrix and $\widehat{\mathbf{W}}_{ij} \geq 0$. More importantly, the approximated matrix $\widehat{\mathbf{W}}$ is automatically normalized and we can find that $\widehat{\mathbf{W}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$. In this case, the Laplacian matrix $\mathbf{L}$ can be considered as $\mathbf{L} = \mathbf{I} - \widehat{\mathbf{W}}$ and we can rewrite $\mathbf{P}$ and $\mathbf{Q}$ as follows:

$$\begin{aligned} \mathbf{P} &= 2\lambda\mathbf{Y} + \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{Z}^T\mathbf{Y} \\ \mathbf{Q} &= \mathbf{Y} + 2\lambda\mathbf{Y}\mathbf{Y}^T\mathbf{Y} \end{aligned} \tag{28}$$

5. Experiments

In the experiments, we verified the performance of the proposed unsupervised HSI classification algorithm on both synthetic datasets and HSI datasets, and then showed several useful analysis. The synthetic benchmark datasets were three sets of data with manifold structure and the HSI datasets are several hyperspectral images (i.e., Salinas, Pavia University, Kennedy Space Center, Samson, Indian Pines, Urban and Japser).

5.1. Experimental Datasets

We conducted experiments on eight widely used hyperspectral datasets:

- Salinas and Salinas-A were acquired by the 224-band AVIRIS sensor over Salinas Valley, California, and characterized by high spatial resolution (3.7-m pixels). Salinas covers 512 lines by 217 samples at as scale of 512 × 217. Salinas ground truth contains 16 classes. Salinas-A is an small subscene of Salinas image and it comprises 86 × 83 pixels located within the same scene at [samples, lines] = [591–676, 158–240] and includes six classes.
- Pavia University is the scene collected by the ROSIS sensor during a flight campaign over Pavia, northern Italy. The number of spectral bands is 103 for Pavia University. Pavia University is a 610 × 610 pixels image, where some pixels contain no information and these samples are discarded. Both hyperspectral image ground truths differentiate nine classes.
- Kennedy Space Center was acquired by the NASA AVIRIS instrument over the Kennedy Space Center (KSC), Florida, on 23 March 1996. They acquired data in 224 bands of 10 nm width with center wavelengths from 400 to 2500 nm and 176 bands were used for the analysis. KSC hyperspectral image contains 512 × 614 pixels. For classification purposes, 13 classes representing the various land cover types that occur in this environment were defined for the site.
- Samson dataset is an image with 95 × 95 pixels and each pixel was recorded at 156 channels covering the wavelengths from 401 nm to 889 nm. The spectral resolution is high up to 3.13 nm and it is not degraded by blank or noisy channels. There are three targets in this image: Soil, Tree and Water.
- Japser Ridge is a hyperspectral image with 100 × 100 pixels. Each pixel was recorded at 224 channels ranging from 380 nm to 2500 nm. The spectral resolution is up to 9.46 nm. There are four end-members latent in these data: Road, Soil, Water and Tree.
- Urban has 210 wavelengths ranging from 400 nm to 2500 nm, resulting in a spectral resolution of 10 nm. There are 307 × 307 pixels, each of which corresponding to a 2 × 2 m^2 area. There are three versions of the ground truth, which contain 4, 5 and 6 end-members respectively, and are introduced in the ground truth.
- Indian Pines was gathered by AVIRIS sensor in northwestern Indiana and consists of 145 × 145 pixels and 224 spectral reflectance bands. The Indian Pines scene contains two-thirds agriculture, and one-third forest or other natural perennial vegetation. The ground truth available is designated into sixteen classes and we reduced the number of bands to 200 by removing bands covering the region of water absorption.

5.2. Evaluation Metrics

In the experiments, we evaluated the clustering results by Purity (P.) and Normalized Mutual Information (NMI).

- P. is the most common metric for clustering results evaluation and it can be formulated as

$$\text{Purity}(\Omega, \widehat{\Omega}) = \frac{1}{n} \sum_i \max_j |\Omega_i \cap \widehat{\Omega}_j| \tag{29}$$

  where Ω is the clustering result set and $\widehat{\Omega}$ is the ground truth. The worst clustering result is very close to 0 and the best clustering result has a purity value equal to 1.
- NMI is a normalization of the mutual information score to scale the results between 0 and 1 as

$$\text{NMI} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{c} n_{i,j} log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i log \frac{n_i}{n})(\sum_{i=1}^{c} \hat{n}_i log \frac{\hat{n}_i}{n})}}, \tag{30}$$

where n_i denotes the number of data contained in the cluster $\mathcal{C}_i (1 \leq i \leq c)$, $\hat{n}_j$ is the number of data belonging to the $\mathcal{L}_j (1 \leq j \leq c)$, and $n_{i,j}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_i$ and the class $\mathcal{L}_j$. The larger is the NMI, the better is the clustering result.

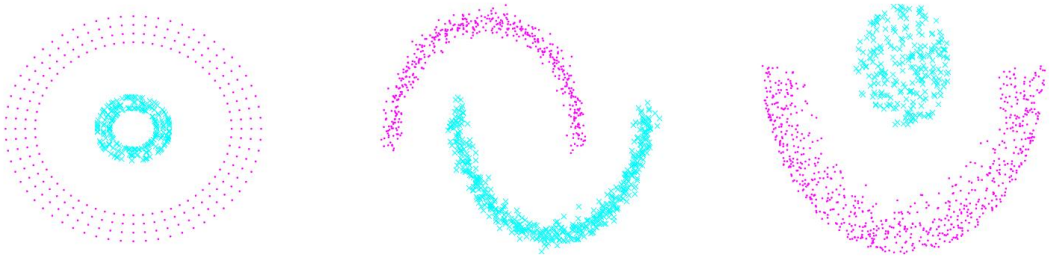
We ran the experiments under the same environment: Intel(R) Core(TM) i7-5930K CPU, 3.50GHz, 64GB memory, Ubuntu 14.04.5 LTS system and Matlab version R2014b. We compared our algorithm with Spectral Clustering (SC), Anchor-based Graph Clustering (AGC), and Nyström Extension Clustering (NEC). The corresponding improved algorithms based on multiplicative update optimization are SC-I, NEC-I, and AGC-I. The affinity matrix of the above algorithms were constructed in three ways and the detailed description of the above affinity matrix is presented in the next section.

5.3. Toy Example

We firstly explored the performance of our algorithm on three synthetic datasets to verify the effectiveness of multiplicative update optimization and two approximated affinity matrix matrices. In this experiment, three synthetic datasets were introduced in our experiment: Cluster in Cluster (CC), Two Spirals (TS), and Crescent Moon (CM). Figure 1 presents the manifold structure of the synthetic datasets in detail. These synthetic datasets contain 2000–40,000 data points that are divided into two groups and they are extremely challenging since clustering algorithms that only consider data point distance have difficulty obtaining a robust result. The algorithms with spectral graph theory provide a more powerful technique in dealing with the manifold information. The resolution for spectral clustering can be divided into two parts: affinity matrix construction and eigenvalue decomposition of the Laplacian matrix. In this paper, we present three formulations for the affinity matrix construction as

$$\begin{aligned}
&\text{Euclidean distance} : \mathbf{W}_{ij} = e^{\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\alpha\hat{\sigma}^2}}, \\
&\text{Nyström extension} : \widehat{\mathbf{W}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}, \quad \mathbf{A}_{ij} = e^{\frac{-||\mathbf{u}_i - \mathbf{u}_j||_2^2}{2\alpha\hat{\sigma}^2}}, \mathbf{B}_{ij} = e^{\frac{-||\mathbf{u}_i - \mathbf{x}_j||_2^2}{2\alpha\hat{\sigma}^2}}, \\
&\text{Anchor-based graph} : \widehat{\mathbf{W}} = \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{Z}^T, \quad \mathbf{Z}_{ij} = \frac{\mathbf{d}_{i,k+1} - \mathbf{d}_{ij}}{k\mathbf{d}_{i,k+1} - \sum_{j'=1}^{k} \mathbf{d}_{ij'}}.
\end{aligned} \tag{31}$$

where $\mathbf{x}$ is the whole sample and $\mathbf{u}$ is the chosen data points. α is the parameter to control the neighbor of data points for Euclidean distance and we set $\alpha = 10$. $\mathbf{A}$ is the affinity matrix for anchors (chosen data points) and $\mathbf{B}$ stores the similarity between anchors (chosen data points) and the remaining ones. $\mathbf{d}_{ij}$ denotes the distance between the ith data point and the jth anchor, which can be considered as chosen data points, and $\mathbf{d}_{i1}, \mathbf{d}_{i2}, ..., \mathbf{d}_{in}$ are ordered from small to large. According to Author1 [20], the parameter k for anchor-based graph was set to 10, which provided a good performance in most cases. Note that the last two affinity matrices are the approximated solution for the original affinity matrix. The sample scale was set to 10, which means we randomly selected one-tenth of data points as the anchors or the chosen data points.



(**a**) Cluster in cluster (CC) (**b**) Two spirals (TS) (**c**) Crescent moon (CM)

Figure 1. The synthetic datasets.

Compared with the traditional eigenvalue decomposition of the Laplacian matrix, we propose a multiplicative update optimization to get a more efficient solution of eigenvalue decomposition. In our experiments, the number of iterations was about 150 and we obtained good results in most cases. Besides the above-mentioned parameters, the other parameters of the compared algorithms and our improved algorithms were tuned to the optimum.

Table 2. Clustering results on synthetic dataset (CC).

	SC			**SC-I**			**NEC**			**NEC-I**			**AGC**			**AGC-I**		
	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT
Num. = 2000	1.00	1.00	1.02	1.00	1.00	0.35	0.68	0.25	0.05	1.00	1.00	0.08	1.00	1.00	0.05	1.00	1.00	0.06
Num. = 4000	1.00	1.00	1.45	1.00	1.00	1.35	0.65	0.09	0.14	1.00	1.00	0.17	1.00	1.00	0.15	1.00	1.00	0.14
Num. = 6000	1.00	1.00	3.12	1.00	1.00	2.94	0.69	0.26	0.36	1.00	1.00	0.48	1.00	1.00	0.26	1.00	1.00	0.29
Num. = 8000	1.00	1.00	5.53	1.00	1.00	5.23	0.67	0.25	0.68	1.00	1.00	0.81	1.00	1.00	0.56	1.00	1.00	0.54
Num. = 10,000	1.00	1.00	9.05	1.00	1.00	7.87	0.68	0.25	0.81	1.00	1.00	1.25	1.00	1.00	0.69	1.00	1.00	0.86
Num. = 12,000	1.00	1.00	13.04	1.00	1.00	11.35	0.50	0.00	1.91	1.00	1.00	1.89	1.00	1.00	0.83	1.00	1.00	1.20
Num. = 14,000	1.00	1.00	18.39	1.00	1.00	15.23	0.57	0.02	2.67	1.00	1.00	2.32	1.00	1.00	1.13	1.00	1.00	1.62
Num. = 16,000	1.00	1.00	23.99	1.00	1.00	21.44	0.52	0.00	3.66	1.00	1.00	3.01	1.00	1.00	1.33	1.00	1.00	2.17
Num. = 18,000	1.00	1.00	31.05	1.00	1.00	25.00	0.63	0.19	3.25	1.00	1.00	4.42	1.00	1.00	1.85	1.00	1.00	2.87
Num. = 20,000	1.00	1.00	39.52	1.00	1.00	31.52	0.69	0.27	6.59	1.00	1.00	4.55	1.00	1.00	2.23	1.00	1.00	4.58
Num. = 22,000	1.00	1.00	50.36	1.00	1.00	43.48	0.50	0.00	5.58	1.00	1.00	7.19	1.00	1.00	3.06	1.00	1.00	5.57
Num. = 24,000	1.00	1.00	62.55	1.00	1.00	52.81	0.54	0.00	7.13	1.00	1.00	8.40	1.00	1.00	3.79	1.00	1.00	6.54
Num. = 26,000	1.00	1.00	76.38	1.00	1.00	60.66	0.53	0.00	9.17	1.00	1.00	8.88	1.00	1.00	4.54	1.00	1.00	7.57
Num. = 28,000	1.00	1.00	93.06	1.00	1.00	70.78	0.69	0.26	11.59	1.00	1.00	12.34	0.83	0.47	5.45	1.00	1.00	8.78
Num. = 30,000	1.00	1.00	111.98	1.00	1.00	81.95	0.74	0.28	19.52	1.00	1.00	14.34	1.00	1.00	8.12	1.00	1.00	10.31
Num. = 32,000	1.00	1.00	182.78	1.00	1.00	95.47	0.59	0.15	23.14	1.00	1.00	15.63	0.83	0.48	10.01	1.00	1.00	12.43
Num. = 34,000	1.00	1.00	212.34	1.00	1.00	96.86	0.63	0.20	17.35	1.00	1.00	18.90	1.00	1.00	10.30	1.00	1.00	13.72
Num. = 36,000	1.00	1.00	277.53	1.00	1.00	104.32	0.51	0.00	21.86	1.00	1.00	19.13	1.00	1.00	31.71	1.00	1.00	14.41
Num. = 38,000	1.00	1.00	348.30	1.00	1.00	115.03	0.50	0.00	24.99	1.00	1.00	22.33	1.00	1.00	23.17	1.00	1.00	16.07
Num. = 40,000	1.00	1.00	475.56	1.00	1.00	138.64	0.57	0.12	43.01	1.00	1.00	24.66	1.00	1.00	18.09	1.00	1.00	17.70
Average	1.00	1.00	101.85	1.00	1.00	49.11	0.60	0.13	10.17	1.00	1.00	8.54	0.98	0.95	6.37	1.00	1.00	6.37

Table 3. Clustering results on synthetic dataset (TS).

	SC			**SC-I**			**NEC**			**NEC-I**			**AGC**			**AGC-I**		
	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT
Num. = 2000	0.97	0.83	1.01	1.00	0.98	0.50	0.50	0.01	0.08	0.99	0.93	0.17	1.00	1.00	0.82	0.95	0.71	0.14
Num. = 4000	0.98	0.85	1.40	0.98	0.88	1.92	0.73	0.22	0.38	1.00	0.96	0.87	1.00	1.00	0.40	0.98	0.87	0.24
Num. = 6000	0.97	0.83	2.95	0.97	0.82	3.98	0.50	0.00	0.68	1.00	0.98	1.76	1.00	1.00	0.87	0.99	0.93	0.43
Num. = 8000	0.97	0.79	5.41	0.99	0.93	7.40	0.50	0.00	1.26	0.99	0.94	2.63	1.00	1.00	1.04	1.00	0.96	0.97
Num. = 10,000	0.97	0.81	8.37	0.99	0.95	11.27	0.50	0.00	2.35	1.00	0.95	4.09	1.00	1.00	2.68	0.87	0.45	1.90
Num. = 12,000	0.97	0.79	13.39	0.99	0.94	15.81	0.50	0.00	3.46	0.99	0.95	7.65	1.00	1.00	2.75	0.95	0.71	2.41
Num. = 14,000	0.97	0.79	18.70	0.98	0.89	20.61	0.71	0.29	5.01	0.99	0.91	11.43	1.00	1.00	4.77	0.98	0.87	3.31
Num. = 16,000	0.97	0.79	26.49	0.83	0.35	29.71	0.51	0.03	6.79	0.96	0.80	18.55	1.00	1.00	5.68	0.99	0.93	4.14
Num. = 18,000	0.97	0.81	32.98	0.99	0.92	34.42	0.68	0.25	8.97	0.99	0.95	20.36	1.00	1.00	6.82	1.00	0.96	4.03
Num. = 20,000	0.97	0.82	43.03	0.99	0.90	43.23	0.50	0.00	10.82	0.99	0.93	23.61	1.00	1.00	6.96	0.92	0.59	5.59
Num. = 22,000	0.97	0.79	55.62	0.99	0.93	52.66	0.72	0.30	15.71	0.99	0.94	32.81	1.00	1.00	11.39	0.95	0.71	5.30
Num. = 24,000	0.97	0.80	72.02	0.99	0.95	63.61	0.52	0.01	16.68	0.99	0.93	34.86	1.00	1.00	10.94	0.98	0.87	6.48
Num. = 26,000	0.97	0.80	85.32	0.99	0.94	72.83	0.53	0.03	21.37	0.99	0.91	48.00	1.00	1.00	10.66	0.99	0.93	7.71
Num. = 28,000	0.97	0.80	102.27	0.99	0.95	83.75	0.50	0.00	25.86	0.99	0.95	52.51	1.00	1.00	11.99	1.00	0.96	8.52
Num. = 30,000	0.97	0.81	149.99	1.00	0.98	97.31	0.51	0.03	32.83	0.99	0.94	64.45	1.00	1.00	17.10	1.00	1.00	9.06
Num. = 32,000	0.97	0.81	190.72	0.99	0.93	118.01	0.50	0.00	38.24	0.99	0.94	72.44	1.00	1.00	18.38	1.00	0.98	11.40
Num. = 34,000	0.97	0.81	258.66	0.98	0.88	128.90	0.51	0.03	47.00	0.99	0.95	71.32	1.00	1.00	24.32	0.91	0.57	12.86
Num. = 36,000	0.97	0.81	358.37	0.98	0.86	137.45	0.50	0.00	51.82	0.99	0.94	83.11	1.00	1.00	30.24	0.98	0.89	13.48
Num. = 38,000	0.97	0.80	459.32	0.97	0.82	160.64	0.50	0.00	57.57	0.68	0.10	94.87	1.00	1.00	20.89	0.98	0.04	15.44
Num. = 40,000	0.97	0.81	636.23	0.99	0.94	201.30	0.50	0.00	67.46	1.00	0.97	115.24	1.00	1.00	30.73	1.00	1.00	15.88
Average	0.97	0.81	126.31	0.98	0.89	64.27	0.55	0.06	20.72	0.98	0.89	38.04	1.00	1.00	10.97	0.97	0.80	6.46

Table 4. Clustering results on synthetic dataset (CM).

	SC			**SC-I**			**NEC**			**NEC-I**			**AGC**			**AGC-I**		
	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT	P.	NMI	CT
Num. = 2000	1.00	1.00	0.38	1.00	0.98	0.70	0.50	0.00	0.09	1.00	1.00	0.17	0.56	0.19	0.86	1.00	1.00	0.08
Num. = 4000	1.00	1.00	1.34	0.99	0.92	2.43	0.50	0.00	1.50	1.00	1.00	0.85	1.00	1.00	0.41	1.00	1.00	0.29
Num. = 6000	1.00	1.00	2.70	0.99	0.90	5.22	0.50	0.00	1.08	1.00	1.00	1.63	1.00	1.00	1.14	1.00	1.00	0.81
Num. = 8000	1.00	1.00	5.71	0.99	0.91	9.64	0.89	0.53	1.90	1.00	1.00	3.22	1.00	1.00	2.13	1.00	1.00	1.48
Num. = 10,000	1.00	1.00	8.46	0.99	0.95	15.39	0.50	0.00	3.05	1.00	1.00	5.32	1.00	1.00	2.20	1.00	1.00	2.34
Num. = 12,000	1.00	1.00	12.55	0.99	0.95	21.76	0.50	0.01	4.70	1.00	1.00	8.24	1.00	1.00	3.73	1.00	1.00	3.42
Num. = 14,000	1.00	1.00	18.24	0.99	0.94	27.11	0.50	0.00	8.47	1.00	1.00	12.49	1.00	1.00	3.74	1.00	1.00	4.64
Num. = 16,000	1.00	1.00	26.76	0.93	0.63	39.33	0.50	0.00	10.85	1.00	1.00	17.26	1.00	1.00	4.50	1.00	1.00	6.28
Num. = 18,000	1.00	1.00	34.21	0.99	0.92	44.15	0.90	0.55	15.68	1.00	1.00	20.80	1.00	1.00	6.51	1.00	1.00	7.63
Num. = 20,000	1.00	1.00	43.86	0.99	0.92	57.08	0.50	0.00	21.38	1.00	1.00	25.60	1.00	1.00	7.78	1.00	1.00	9.60
Num. = 22,000	1.00	1.00	55.55	0.99	0.94	72.23	0.50	0.00	27.26	1.00	1.00	33.16	1.00	1.00	8.48	1.00	1.00	11.20
Num. = 24,000	1.00	1.00	69.45	0.99	0.95	86.58	0.68	0.25	27.87	1.00	1.00	38.49	1.00	1.00	8.98	1.00	1.00	13.34
Num. = 26,000	1.00	1.00	101.07	0.99	0.95	99.36	0.50	0.01	48.77	1.00	1.00	44.62	1.00	1.00	11.41	1.00	1.00	15.60
Num. = 28,000	1.00	1.00	114.92	0.99	0.95	114.37	0.50	0.00	39.56	1.00	1.00	55.30	1.00	1.00	11.83	1.00	1.00	17.92
Num. = 30,000	1.00	1.00	149.53	1.00	0.96	136.30	0.91	0.59	76.80	1.00	1.00	63.81	1.00	1.00	14.98	1.00	1.00	21.26
Num. = 32,000	1.00	1.00	209.87	0.99	0.93	158.11	0.85	0.49	84.71	1.00	1.00	67.09	1.00	1.00	16.84	1.00	1.00	27.18
Num. = 34,000	1.00	1.00	270.50	0.99	0.91	167.20	0.50	0.00	82.73	1.00	1.00	79.64	1.00	1.00	20.62	1.00	1.00	28.29
Num. = 36,000	1.00	1.00	381.08	0.99	0.91	181.04	0.62	0.04	76.13	1.00	1.00	89.44	1.00	1.00	20.62	1.00	1.00	30.00
Num. = 38,000	1.00	1.00	594.83	0.96	0.75	223.33	0.77	0.37	99.99	1.00	1.00	100.11	1.00	1.00	21.47	1.00	1.00	33.28
Num. = 40,000	1.00	1.00	754.12	0.99	0.93	258.65	0.50	0.00	108.56	1.00	1.00	119.93	1.00	1.00	32.86	1.00	1.00	36.46
Average	1.00	1.00	142.76	0.99	0.91	86.00	0.61	0.14	37.05	1.00	1.00	39.36	0.98	0.96	10.05	1.00	1.00	13.55

Table 2–4 present the performance of the above six methods on three synthetic datasets. SC and SC-I provided a good clustering result since the corresponding affinity matrix considered the similarity of the whole data points; however, these two methods also needed more time to calculate the Euclidean distance among samples. Note that the proposed multiplicative update algorithm delivered a substantial efficiency increase, taking only half the time to get a similar clustering result. NEC and AGC had the benefit of the approximated affinity matrix and took only about one-tenth the time, but NEC was not robust enough to get a stable resolution of the eigenvalue decomposition. Compared with NEC, the improved algorithm NEC-I provided a better clustering result because of the orthonormal constraint and nonnegative constraint. AGC performed better than SC and NEC in terms of effectiveness and efficiency in the experiments, as it utilized the anchor-based affinity matrix, and the proposed AGC-I also had a good performance.

5.4. HSI Clustering Analysis

In this section, a further study is presented to illustrate the performance of the proposed multiplicative update algorithm and the efficiency of the approximated affinity matrix mentioned in Section 4 on several popular hyperspectral image datasets. We followed the experiment setting in the previous section where the parameter α was set to 10 and the parameter k was set to 10. In addition, the parameter λ was set to 0.5 and the other parameters were tuned to the optimum for fair competition. Note that the affinity matrix for the hyperspectral image datasets was different from the previous section because it needed to consider both the brightness value and the spatial information. In this case, the affinity matrix $\mathbf{W}$ can be rewritten as

$$\mathbf{W}_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\alpha\bar{\sigma}_x^2} - \frac{||\mathbf{l}_i - \mathbf{l}_j||_2^2}{2\alpha\bar{\sigma}_l^2}}, \tag{32}$$

where $\mathbf{l}$ is the pixel location and the parameter α was set to 10 for both the brightness value and the spatial information. The affinity matrices $\mathbf{A}$ and $\mathbf{B}$ for NEC were constructed in the same way. Meanwhile, The affinity matrix for AGC is provided as

$$\begin{aligned} \widehat{\mathbf{W}} &= \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{Z}^T, \\ \mathbf{Z}_{ij} &= \frac{\mathbf{d}_{i,k+1} - \mathbf{d}_{ij}}{k\mathbf{d}_{i,k+1} - \sum_{j'=1}^{k}\mathbf{d}_{ij'}}, \end{aligned} \tag{33}$$

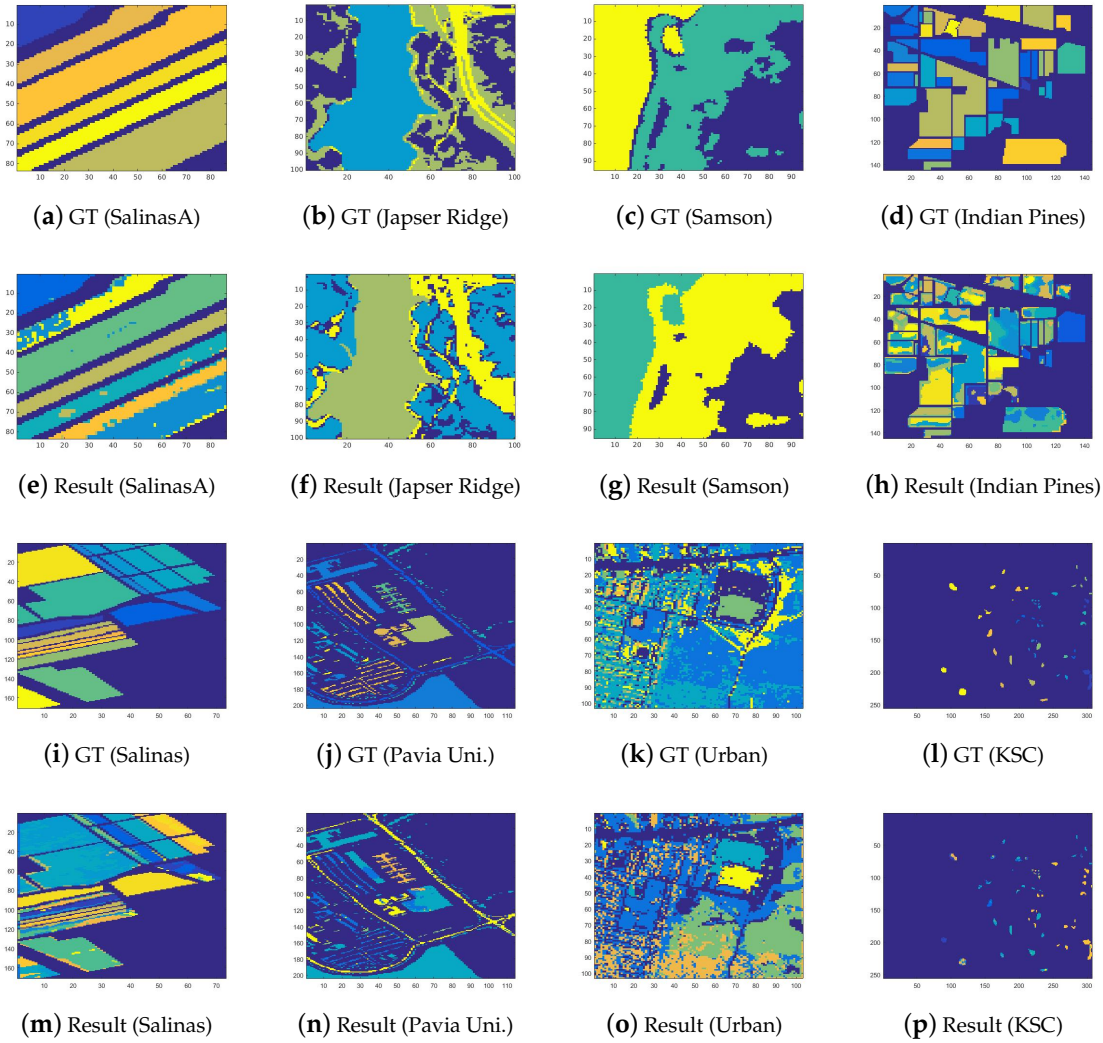
where $\mathbf{d}_{ij} = ||\mathbf{x}_i - \mathbf{u}_j||_2^2 + ||\bar{\mathbf{x}}_i - \mathbf{u}_j||_2^2$ and $\bar{\mathbf{x}}$ is the mean of the brightness value around pixel $\mathbf{x}$.

Figure 2 and Table 5 present the experimental results, which were evaluated by Purity and NMI on the hyperspectral image datasets. We made the following observations:

- SC and the corresponding improved algorithm SC-I achieved competitive performance in term of Purity and NMI. However, SC took more time solving eigenvalue decomposition of Laplacian matrix and our improved algorithm provided a more efficient solution because of the utilization of the multiplicative update optimization. Meanwhile, it took more time to process India Pines because of the rapid growth of time complexity of eigenvalue decomposition of Laplacian matrix caused by the increase of spatial resolution and classes. Note that SC-I, which is based on the multiplicative update algorithm, slightly outperformed SC in terms of Purity and NMI, illustrating that the nonnegative constraint and the orthonormal constraint provided a better indicator matrix. This made it easier to get a robust clustering result by the later processing, such as k-means.
- NEC and AGC are two efficient improved algorithms and they took only one-twentieth the time in our experiments. Moreover, NEC and AGC could be used on large-scale hyperspectral image datasets such as KSC and Urban, while SC ran out of memory in dealing with the above large-scale datasets because of the storage and time complexity of the affinity matrix. However, the experimental results also illustrate that NEC was not robust enough, which might be because

the affinity matrix **A** can be indefinite and the inverse matrix contains plural elements, making it difficult to get a robust clustering result by k-means. Besides NEC, the other methods did not struggle with this problem, and also provided a better performance than NEC.

- The proposed NEC-I and AGC-I outperformed the other methods in terms of effectiveness and efficiency. NEC-I and AGC-I firstly take the advantage of sample techniques including Nyström extension and anchor-based graph, which allow them to be used on large-scale hyperspectral image datasets. Furthermore, the proposed multiplicative update algorithm provided an efficient resolution for eigenvalue decomposition of Laplacian matrix. The results presented in Table 5 illustrate that NEC-I and AGC-I performed better than NEC and AGC in most cases. The proposed multiplicative update optimization is flexible and well-knit with the approximated affinity matrix such as Nyström extension and anchor-based graph.



(**a**) GT (SalinasA) (**b**) GT (Japser Ridge) (**c**) GT (Samson) (**d**) GT (Indian Pines)

(**e**) Result (SalinasA) (**f**) Result (Japser Ridge) (**g**) Result (Samson) (**h**) Result (Indian Pines)

(**i**) GT (Salinas) (**j**) GT (Pavia Uni.) (**k**) GT (Urban) (**l**) GT (KSC)

(**m**) Result (Salinas) (**n**) Result (Pavia Uni.) (**o**) Result (Urban) (**p**) Result (KSC)

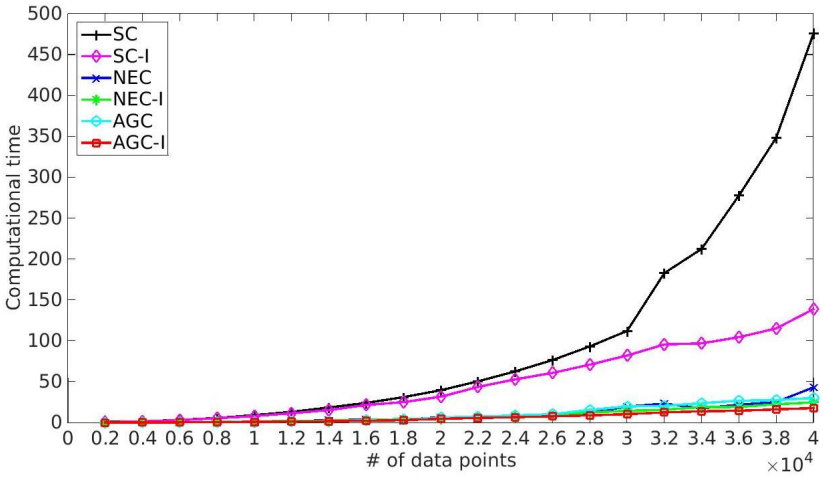
Figure 2. HSI ground truth and results.

Table 5. Clustering results on hyperspectral image datasets. The bold numbers represent the best results in terms of purity, normalization of the mutual information and computational time.

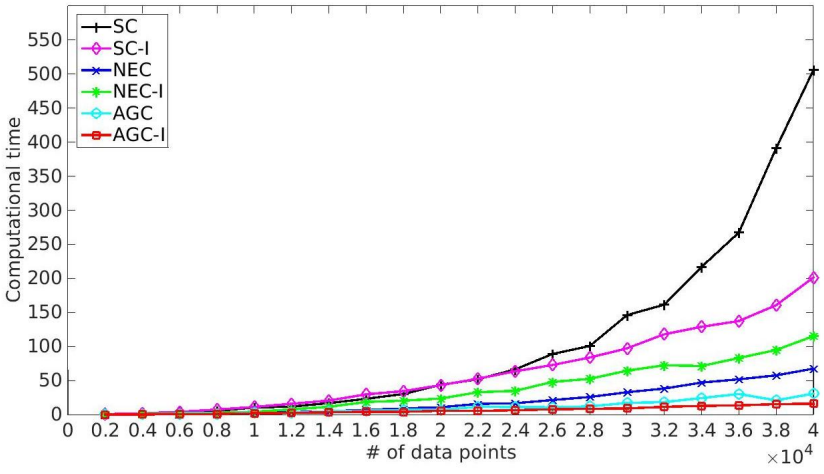
	SC			**SC-I**			**NEC**			**NEC-I**			**AGC**			**AGC-I**		
	PUI.	NMI	CT	PUI.	NMI	CT	PUI.	NMI	CT	PUI.	NMI	CT	PUI.	NMI	CT	PUI.	NMI	CT
Samson	0.85	0.61	6.57	0.85	0.60	5.77	0.73	0.53	**0.10**	0.85	0.60	0.17	0.88	0.73	0.19	**0.91**	**0.75**	0.19
Jasper	0.83	0.71	10.31	**0.91**	**0.76**	6.43	0.70	0.56	**0.03**	0.83	0.71	0.11	0.72	0.66	0.09	0.82	0.70	0.14
SalinasA	0.81	0.80	4.77	**0.85**	0.79	4.31	0.78	0.77	**0.06**	0.80	0.81	0.17	0.79	0.78	0.10	0.84	**0.81**	0.15
India Pines	0.36	0.44	66.21	**0.46**	0.46	45.37	0.43	0.45	**0.53**	0.43	**0.49**	1.29	0.35	0.43	0.58	0.42	0.46	1.46
Salinas	-	-	-	-	-	-	0.60	**0.72**	**1.62**	**0.62**	0.71	4.62	0.56	0.67	2.44	0.56	0.71	3.55
Pavia Uni.	-	-	-	-	-	-	0.47	0.34	**1.34**	**0.61**	**0.57**	3.34	0.46	0.51	3.40	0.54	**0.57**	3.67
KSC	-	-	-	-	-	-	0.46	**0.57**	**1.16**	**0.51**	0.52	5.97	0.47	0.52	6.10	**0.51**	0.53	6.48
Urban	-	-	-	-	-	-	0.40	0.12	**0.41**	0.45	0.21	3.01	**0.51**	**0.33**	1.14	0.50	0.29	3.12
Average	0.68	0.58	27.69	**0.74**	**0.60**	**19.19**	0.57	0.51	**0.66**	**0.64**	0.58	2.34	0.59	0.58	1.76	**0.64**	**0.60**	2.35

5.5. Computational Time

Figure 3 lists the computational time on three synthetic datasets. We ran the experiments under the same environment: Intel(R) Core(TM) i7-5930K CPU, 3.50GHz, 64GB memory, Ubuntu 14.04.5 LTS system and Matlab version R2014b. The methods listed in Figure 3 achieved similar clustering results when there were fewer than 10,000 data points, and SC and SC-I took more time than the other methods when there were more than 10,000 data points. Moreover, the computational time grew rapidly along with the increase of the number of data. The proposed improved algorithm SC-I took only about half the time with more than 30,000 data points. Compared with the above two methods, NEC, AGC and the corresponding improved algorithms NEC-I and AGC-I provided better performance in terms of computational time. Meanwhile, the affinity matrix constructed by the anchor-based graph was better than the affinity matrix constructed by Nyström extension, as the anchor-based graph provided a better way to measure the similarity of data points.

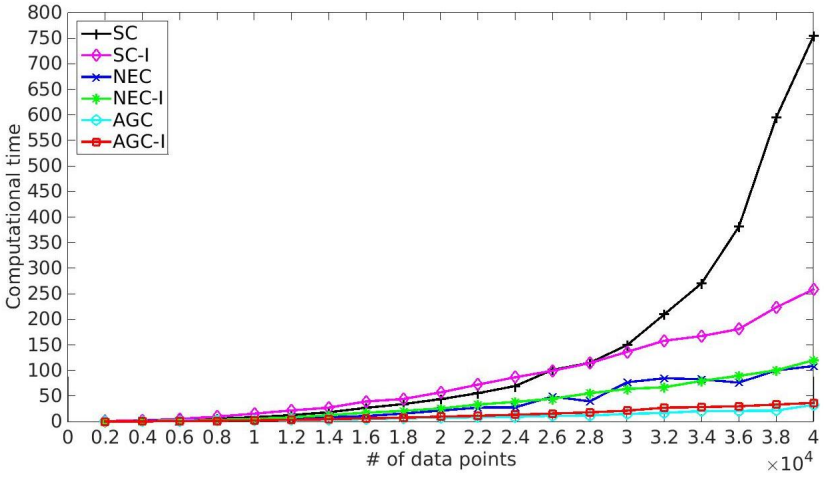


(**a**) Computational time on the synthetic dataset (CC)



(**b**) Computational time on the synthetic dataset (TS)

Figure 3. *Cont.*



(**c**) Computational time on the synthetic dataset (CM)

Figure 3. Computational time on three synthetic datasets.

6. Conclusions

In this paper, we briefly review the classical spectral clustering technique for unsupervised HSI classification, and two major problems in dealing with large-scale HSI datasets, namely affinity matrix construction and eigenvalue decomposition of Laplacian matrix. Firstly, we introduce two efficient affinity matrix approximation methods, namely Nyström extension and anchor-based graph, by sampling several HSI pixels. Furthermore, we propose an efficient and effective multiplicative update algorithm to get a robust resolution of eigenvalue decomposition and the experimental results also illustrate that the combination of the affinity matrix approximation and the multiplicative update optimization outperformed the other methods. More importantly, the proposed improved algorithm provides an efficient solution for large-scale HSI classification where the traditional spectral clustering have no capability to deal with them.

Author Contributions: All authors made significant contributions to the manuscript. Y.Z., Y.Y. and Q.W. conceived the research and designed the research framework; Y.Z. designed and implemented the algorithm; and Y.Y. and Q.W. analyzed the results and verified the theory. All authors contributed to the editing of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grants U1864204 and 61773316; State Key Program of National Natural Science Foundation of China under Grant 61632018; Natural Science Foundation of Shaanxi Province under Grant 2018KJXX-024; Projects of Special Zone for National Defense Science and Technology Innovation; Fundamental Research Funds for the Central Universities under Grant 3102017AX010; and Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ma, D.; Yuan, Y.; Wang, Q. Hyperspectral anomaly detection via discriminative feature learning with multiple-dictionary sparse representation. *Remote Sens.* **2018**, *10*, 745.
2. Wang, Q.; He, X.; Li, X. Locality and structure regularized low rank representation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, doi:10.1109/TGRS.2018.2862899.
3. He, X.; Wang, Q.; Li, X. Spectral-spatial Hyperspectral image classification via locality and structure constrained low-rank representation. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 23–27 July 2018.
4. Wang, Q.; Meng, Z.; Li, X. Locality adaptive discriminant analysis for spectral-spatial classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2077–2081.
5. Yuan, Y.; Lin, J.; Wang, Q. Hyperspectral image classification via multi-task joint sparse representation and stepwise mrf optimization. *IEEE Trans. Cybern.* **2016**, *46*, 2966–2977.

6. Wang, Q.; Zhang, F.; Li, X. Optimal clustering framework for hyperspectral band selection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5910–5922.
7. Wang, Q.; Wan, J.; Li, X. Robust Hierarchical Deep Learning for Vehicular Management. *IEEE Trans. Veh. Technol.* **2018**, doi:10.1109/TVT.2018.2883046.
8. Wang, Q.; Chen, M.; Nie, F.; Li, X. Detecting Coherent Groups in Crowd Scenes by Multiview Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, doi:10.1109/TPAMI.2018.2875002.
9. Yan, Q.; Ding, Y.; Xia, Y.; Chong, Y.; Zheng, C. Class probability propagation of supervised information based on sparse subspace clustering for hyperspectral images. *Remote Sens.* **2017**, *9*, 1017.
10. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of vhr remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167.
11. Xie, H.; Zhao, A.; Huang, S.; Han, J.; Liu, S.; Xu, X.; Luo, X.; Pan, H.; Du, Q.; Tong, X. Unsupervised hyperspectral remote sensing image clustering based on adaptive density. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 632–636.
12. Chen, M.; Wang, Q.; Li, X. Discriminant analysis with graph learning for hyperspectral image classification. *Remote Sens.* **2018**, *10*, 836.
13. Zhang, H.; Zhai, H.; Zhang, L.; Li, P. Spectral-spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3672–3684.
14. Zhang, L.; You, J. A spectral clustering based method for hyperspectral urban image. In Proceedings of the 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, 6–8 March 2017; pp. 1–3.
15. Matasci, G.; Volpi, M.; Kanevski, M.; Bruzzone, L.; Tuia, D. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3550–3564.
16. Crawford, M.M.; Tuia, D.; Yang, H.L. Active learning: Any value for classification of remotely sensed data? *Proc. IEEE* **2013**, *101*, 593–608.
17. Tuia, D.; Persello, C.; Bruzzone, L. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 41–57.
18. Guo, X.; Huang, X.; Zhang, L.; Zhang, L.; Plaza, A.; Benediktsson, J.A. Support tensor machines for classification of hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3248–3264.
19. Wang, R.; Nie, F.; Yu, W. Fast spectral clustering with anchor graph for large hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2003–2007.
20. Nie, F.; Wang, X.; Jordan, M.I.; Huang, H. The constrained laplacian rank algorithm for graph-based clustering. In Proceeding of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
21. Bezdek, J.C. Pattern recognition with fuzzy objective function algorithms. *Adv. Appl. Pattern Recognit.* **1981**, *22*, 203–239.
22. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496.
23. Buckley, J.J. Fuzzy hierarchical analysis. *Fuzzy Sets Syst.* **1985**, *17*, 233–247.
24. Vijendra, S. Efficient clustering for high dimensional data: Subspace based clustering and density based clustering. *Inf. Technol. J.* **2011**, *10*, 1092–1105.
25. Zhong, Y.; Zhang, L.; Gong, W. Unsupervised remote sensing image classification using an artificial immune network. *Int. J. Remote Sens.* **2011**, *32*, 5461–5483.
26. Zhong, Y.; Zhang, S.; Zhang, L. Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2290–2301.
27. Zhao, Y.; Yuan, Y.; Nie, F.; Wang, Q. Spectral clustering based on iterative optimization for large-scale and high-dimensional data. *Neurocomputing* **2018**, *318*, 227–235.
28. Belongie, S.; Fowlkes, C.; Chung, F.; Malik, J. Spectral partitioning with indefinite kernels using the Nyström extension. In Proceeding of the European Conference on Computer Vision,Copenhagen, Denmark, 28–31 May 2002; pp. 531–542.
29. Fowlkes, C.; Belongie, S.; Chung, F.; Malik, J. Spectral grouping using the Nystrom method. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 214–225.
30. Zhu, W.; Nie, F.; Li, X. Fast Spectral Clustering with efficient large graph construction. In Proceedings of the IEEE International Conference on Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 2492–2496.

31. Nie, F.; Zhu, W.; Li, X. Unsupervised Large Graph Embedding. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2422–2428.
32. Bai, J.; Xiang, S.; Pan, C. A graph-based classification method for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 803–817.
33. Camps-Valls, G.; Marsheva, T.V.B.; Zhou, D. Semi-supervised graph-based hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3044–3054.
34. Yokoya, N.; Yairi, T.; Iwasaki, A. Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 528–537.
35. Jia, S.; Qian, Y. Constrained nonnegative matrix factorization for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 161–173.
36. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814.
37. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790.
38. Rodarmel, C.; Shan, J. Principal component analysis for hyperspectral image classification. *Surv. Land Inf. Sci.* **2002**, *62*, 115–122.
39. Wang, Q.; Lin, J.; Yuan, Y. Salient band selection for hyperspectral image classification via manifold ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1279–1289.
40. Fowlkes, C.; Belongie, S.; Malik, J. Efficient spatiotemporal grouping using the Nyström method. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
41. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
42. Türkmen, A.C. A review of nonnegative matrix factorization methods for clustering. *Comput. Sci.* **2015**, *1*, 405–408.
43. Liu, W.; He, J.; Chang, S.F. Large Graph Construction for Scalable Semi-Supervised Learning. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 22–24 June 2010.
44. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4085–4098.
45. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985.
46. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107.
47. Nie, F.; Ding, C.; Luo, D.; Huang, H. Improved minmax cut graph clustering with nonnegative relaxation. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain, 20–24 September 2010; pp. 451–466.
48. Tang, X.; Jiao, L.; Emery, W.J.; Liu, F.; Zhang, D. Two-stage reranking for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5798–5817.
49. Zhang, X.; Jiao, L.; Liu, F.; Bo, L.; Gong, M. Spectral clustering ensemble applied to sar image segmentation. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2126–2136.
50. Wang, Q.; Wan, J.; Nie, F; Liu, B.; Yan, C.; Li, X. Hierarchical Feature Selection for Random Projection. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, doi:10.1109/TNNLS.2018.2868836.
51. Wang, Q.; Qin, Z.; Nie, F.; Li, X. Spectral Embedded Adaptive Neighbors Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, doi:10.1109/TNNLS.2018.2861209.