Applied Soft Computing Journal 73 (2018) 727–734

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Sequential spectral clustering of hyperspectral remote sensing image over bipartite graph

Aidin Hassanzadeh [a,*], Arto Kaarna [a], Tuomo Kauranne [b]

[a] Machine Vision and Pattern Recognition Laboratory, School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland
[b] Mathematics Laboratory, School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland[1]

HIGHLIGHTS

- A sequential Spectral Clustering for remote sensing hyperspectral images.
- A bipartite graph representation to reduce the time-space complexity of affinity matrix.
- Mini-batch K-means to speed up large-scale hyperspectral image clustering.
- The proposed sequential spectral clustering outperforms benchmark clustering algorithms.

ARTICLE INFO

Article history:
Received 1 December 2017
Received in revised form 26 June 2018
Accepted 11 September 2018
Available online xxxx

Keywords:
Hyperspectral image
Spectral clustering
Bipartite graph
Anchor graph
Mini-batch K-means

ABSTRACT

Unsupervised classification is a crucial step in remote sensing hyperspectral image analysis where producing labeled data is a laborious task. Spectral Clustering is an appealing graph-partitioning technique with outstanding performance on data with non-linear dependencies. However, Spectral Clustering is restricted to small-scale data and neither has been effectively applied to hyperspectral image analysis. In this paper, the unsupervised classification of hyperspectral images is addressed through a sequential spectral clustering that can be extended to the large-scale hyperspectral image. To this end, this paper utilizes a bipartite graph representation along with a sequential singular value decomposition and mini-batch K-means for unsupervised classification of hyperspectral imagery. We evaluate the proposed algorithm with several benchmark hyperspectral datasets including *Botswana*, *Salinas*, *Indian Pines*, *Pavia Center Scene* and *Pavia University Scene*. The experimental results show significant improvements made by the proposed algorithm compared to the state-of-art clustering algorithms.

1. Introduction

Unsupervised classification (clustering) is an indispensable technique in several advanced data analysis tasks such as image segmentation, pattern recognition, and data mining. Indeed, clustering plays a significant role in the processing of hyperspectral remote sensing imagery, where labeled samples are laborious to produce or are often inadequate.

A hyperspectral image (HSI) consists of hundreds of contiguous spectral bands that provide detailed discriminative features. An extensive range of spectral information can help to distinguish spectrally similar materials and makes HSI a powerful tool in the paradigm of remote sensing analysis. However, having several fine spectral bands introduces complexities comprised by the *curse of dimensionality* [1,2]. As HSI undergoes with non-linear scattering patterns, variations in the local geometry of the sun-canopy-sensor triangle and nonuniform pixel composition [3], it may usually come with non-linear dependencies. The intrinsic non-linearities in HSI introduce several complexities that can be problematic with traditional clustering algorithms such as K-means.

The spectral clustering (SC) [4,5] is a favorable graph-based clustering approach on eigenvalue decomposition of a data affinity matrix that can capture a broad range of data clusters with different geometric structures. SC is built on a sparse graph representation of the data of interest, where each node in the graph represents an entity in data space and the graph edges represent the respective affinities. Commonly, the m eigenvectors corresponding with the m smallest eigenvalues of the graph Laplacian matrix are used to cluster the entities into discrete clusters of high similarity.

There are neither explicit nor implicit assumptions about data cluster shapes in SC. This makes SC a promising candidate for clustering data with potential non-linearities. SC is an appealing clustering algorithm, but it does not fit well with large sample size datasets. In its construction, it heavily depends on building

* Corresponding author.
E-mail addresses: aidin.hassanzadeh@lut.fi (A. Hassanzadeh), arto.kaarna@lut.fi (A. Kaarna), tuomo.kauranne@lut.fi (T. Kauranne).
[1] LUT School of Engineering Science P.O. Box 20 FI-53851 Lappeenranta Finland

https://doi.org/10.1016/j.asoc.2018.09.015
1568-4946/

pairwise affinity matrix which involves with high memory requirements and thus is inefficient even with data containing a moderate number of samples.

The challenges confronting SC in large-scale applications are three-fold. First and the foremost, the computation of Laplacian matrix of a massive data graph is highly demanding. In particular, the time and space complexity of constructing a Laplacian matrix is $\mathcal{O}(n^2d)$, where n is the number of samples, and d is the dimension of data entries. This quadratic complexity introduces serious scalability problems with the existing processing frameworks that so make SC infeasible when dealing with large-scale applications likely seen in HSI clustering task. Second, eigenvalue decomposition in SC with a very large Laplacian matrix can be very computationally involved. The time and space complexity of the well-known eigenvalue decomposition solvers Lanczos [6] and pre-conditioned conjugate gradient (CG-based) [7] are $\mathcal{O}(in^2m)$ and $\mathcal{O}(n^2m)$ respectively, where m is the number of principal eigenvectors and i is the number of iterations. Third, the traditional SC relies on the Loyd's classic K-means algorithm [8] to partition data to discrete groups. Even though K-means is efficient with large datasets, it cannot reasonably meet the requirements when clustering results are quickly needed or regularly called as an intermediate subroutine.

As a result, SC has not generally been applied to analysis of hyperspectral remote sensing imaging data and even the existing approaches in this context are usually limited only to HSI data containing a few numbers of samples and neither can be applied to large-scale dataset [9–11].

There are several extensions of SC for large-scale applications. [12] proposes a fast and scalable spectral clustering algorithm called the sequential matrix compression (SMC) method. In this algorithm, the computational complexity of SC is addressed by reducing the dimensionality of the Laplacian matrix. [13] implements a sequential spectral clustering algorithm on bipartite graph realization of data.

Not many extensions of SC have been utilized for HSI clustering tasks. [9,11] propose two multi-manifold spectral clustering algorithms on tangential similarities for HSI, but the experiments are restricted to hypercubes of a small number of samples. [10] proposes a spatial–spectral co-clustering clustering algorithm based on bipartite graph representation data. This co-spectral clustering is shown efficient with HSI, but it will be limited to HSI data of a few thousands samples. [14] introduces a fast SC with anchor graphs for HSI clustering where data–anchor points affinities are obtained through local regularization of combined spatial–spectral features. Although the fast SC algorithm has been shown promising in HSI clustering, it still involves several complexities particularly in the computation of affinity matrix that can be slow with real-world application. We also discovered a simple nearest-neighbor affinity metric only based on spectral features can outperform the proposed formulation of spatial–spectral affinity in [14].

In this paper, we present a new algorithm from the family of spectral clustering with low time and space requirements needed for large-scale HSI applications. Inspired by the work in [13], the proposed algorithm follows a sequential procedure that can be effectively extended to large-scale applications of remote sensing HSI. We utilize a bipartite graph representation on radial basis function (Gaussian) kernel to capture data affinities. By this mean, the attempt is to reduce the computational burdens involved with building huge graph Laplacians. To address the complexities present in eigendecomposition, the sequential Singular Value Decomposition (SVD) is used. To compute anchor points in the bipartite graph and the final data cluster assignment, we utilize the mini-batch K-means [15] to meet. We evaluate the performance of the proposed algorithm with five public benchmark HSI data, namely *Botswana*, *Salinas*, *Indian Pines*, *Pavia Center Scene* and *Pavia University Scene*. The experimental results show the efficiency of the proposed SC while applied to HSI unsupervised classification and demonstrate its competence compared to the state of art clustering algorithms.

The remainder of the paper is organized as follows. Section 2 reviews the standard Spectral Clustering algorithm. Section 3 presents the overall procedure of the proposed sequential spectral clustering, presenting the main steps involved. Section 4 presents experimental results and evaluation of the effectiveness of the proposed algorithm. Finally, Section 5 concludes the paper.

2. Spectral clustering

This section presents the main steps involved with the sequential spectral clustering (SSC) algorithm. We first briefly review the standard form of SC and then cover the methodologies used for SSC.

2.0.1. Spectral clustering

Spectral Clustering is defined based on a graph based representation where clustering is performed as partitioning of a similarity graph. Given a set of data points sampled from the original d-dimensional input space $X = \{\mathbf{x}_i\}_{i=1}^n$, SC aims to cluster the data points to K different disjoint groups. SC assumes that the data reside on an undirected weighted graph $G(V, E)$ where $V = \{v_i\}_{i=1}^n$ is the set of nodes representing the data points and $E = \{e_{ij}\}$ is a set of edges connecting the nodes in local proximity. The graph G is commonly constructed by k-nearest neighbor (k-NN) proximity graph where a pair of nodes are connected if either of them appears in the others local neighborhood.

The structure of graph is captured by a weighted affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ that its ijth element $[\mathbf{A}]_{ij}$ describes the strength of similarity between the graph node v_i and the graph node v_j. Given a pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$, let $\mathcal{N}_k(\mathbf{x}_i)$ and $\mathcal{N}_k(\mathbf{x}_j)$ be their k-nearest neighbors sets respectively. An entry in affinity matrix can be defined using a Radial Basis Function (RBF) or Gaussian kernel as follows:

$$[\mathbf{A}]_{ij} = \begin{cases} \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), & \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where γ is a meta parameter that scales the kernel width.

Let $\mathbf{D} \in \mathbb{R}^{n\times n}$ be the graph degree matrix whose diagonal elements are columns sum of affinity matrix $[\mathbf{D}]_{ii} = \sum_j [\mathbf{A}]_{ij}$. SC clusters data points based on the eigenvectors of the graph Laplacian. The graph normalized Laplacian matrix is defined as follows:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \quad (2)$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

The normalized SC formulates the clustering problem through the following optimization problem:

$$\mathbf{G}^* = \arg\min_{\mathbf{G}} trace(\mathbf{G}^T\mathbf{L}\mathbf{G}), \quad (3)$$

where $\mathbf{G}$ is the class indicator matrix and $\mathbf{G}^T\mathbf{G} = \mathbf{I}$.

The objective function in Eq. (3) can be solved through the general eigenvalue problem and that follows $\mathbf{G}^*$ as the m eigenvectors of the Laplacian matrix $\mathbf{L}$ corresponding to its m-largest eigenvalues.

Let each row of the matrix $\mathbf{G}^*$ be a projected point in $\mathbb{R}^m$, the end data clusters are obtained by applying K-means to projected data points. This leads to creating K non-overlapping sets of input data samples.