Pattern Recognition 73 (2018) 247–258

Contents lists available at ScienceDirect

Pattern Recognition

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

Multi-view low-rank sparse subspace clustering

Maria Brbić, Ivica Kopriva*

Laboratory for Machine Learning and Knowledge Representation, Division of Electronics, Rudjer Boskovic Institute, Bijenicka cesta 54, 10000, Zagreb, Croatia

ARTICLE INFO

Article history:
Received 13 March 2017
Revised 10 July 2017
Accepted 23 August 2017
Available online 24 August 2017

Keywords:
Subspace clustering
Multi-view data
Low-rank
Sparsity
Alternating direction method of multipliers
Reproducing kernel Hilbert space

ABSTRACT

Most existing approaches address multi-view subspace clustering problem by constructing the affinity matrix on each view separately and afterwards propose how to extend spectral clustering algorithm to handle multi-view data. This paper presents an approach to multi-view subspace clustering that learns a joint subspace representation by constructing affinity matrix shared among all views. Relying on the importance of both low-rank and sparsity constraints in the construction of the affinity matrix, we introduce the objective that balances between the agreement across different views, while at the same time encourages sparsity and low-rankness of the solution. Related low-rank and sparsity constrained optimization problem is for each view solved using the alternating direction method of multipliers. Furthermore, we extend our approach to cluster data drawn from nonlinear subspaces by solving the corresponding problem in a reproducing kernel Hilbert space. The proposed algorithm outperforms state-of-the-art multi-view subspace clustering algorithms on one synthetic and four real-world datasets.

1. Introduction

In many real-world machine learning problems the same data is comprised of several different representations or views. For example, same documents may be available in multiple languages [1] or different descriptors can be constructed from the same images [2]. Although each of these individual views may be sufficient to perform a learning task, integrating complementary information from different views can reduce the complexity of a given task [3]. Multi-view clustering seeks to partition data points based on multiple representations by assuming that the same cluster structure is shared across views. By combining information from different views, multi-view clustering algorithms attempt to achieve more accurate cluster assignments than one can get by simply concatenating features from different views.

In practice, high-dimensional data often reside in a low-dimensional subspace. When all data points lie in a single subspace, the problem can be set as finding a basis of a subspace and a low-dimensional representation of data points. Depending on the constraints imposed on the low-dimensional representation, this problem can be solved using e.g. Principal Component Analysis (PCA) [4], Independent Component Analysis (ICA) [5] or Non-negative Matrix Factorization (NMF) [6–8]. On the other hand, data points can be drawn from different sources and lie in a union of subspaces. By assigning each subspace to one cluster, one can solve the problem by applying standard clustering algorithms, such as k-means [9]. However, these algorithms are based on the assumption that data points are distributed around centroid and often do not perform well in the cases when data points in a subspace are arbitrarily distributed. For example, two points can have a small distance and lie in different subspaces or can be far and still lie in the same subspace [10]. Therefore, methods that rely on a spatial proximity of data points often fail to provide a satisfactory solution. This has motivated the development of subspace clustering algorithms [10]. The goal of subspace clustering is to identify the low-dimensional subspaces and find the cluster membership of data points. Spectral based methods [11–13] present one approach to subspace clustering problem. They have gained a lot of attention in the recent years due to the competitive results they achieve on arbitrarily shaped clusters and their well defined mathematical principles. These methods are based on the spectral graph theory and represent data points as nodes in a weighted graph. The clustering problem is then solved as a relaxation of the min-cut problem on a graph [14].

One of the main challenges in spectral based methods is the construction of the affinity matrix whose elements define the similarity between data points. Sparse subspace clustering [15] and low-rank subspace clustering [16–19] are among most effective methods that solve this problem. These methods rely on the self-expressiveness property of the data by representing each data point as a linear combination of other data points. Low-Rank Representation (LRR) [16,17] imposes low-rank constraint on the data

* Corresponding author.
E-mail addresses: maria.brbic@irb.hr (M. Brbić), ivica.kopriva@irb.hr, ikopriva@gmail.com, ikopriva@irb.hr (I. Kopriva).

http://dx.doi.org/10.1016/j.patcog.2017.08.024

representation matrix and captures global structure of the data. Low-rank implies that data matrix is represented by a sum of small number of outer products of left and right singular vectors weighted by corresponding singular values. Under assumption that subspaces are independent and data sampling is sufficient, LRR guarantees exact clustering. However, for many real-world datasets this assumption is overly restrictive and the assumption that data is drawn from disjoint subspaces would be more appropriate [20,21]. On the other hand, Sparse Subspace Clustering (SSC) [15] represents each data point as a sparse linear combination of other points and captures local structure of the data. Learning representation matrix in SSC can be interpreted as sparse coding [22–27]. However, compared to sparse coding where dictionary is learned such that the representation is sparse [28,29], SSC is based on self-representation property i.e. data matrix stands for a dictionary. SSC also succeeds when data is drawn from independent subspaces and the conditions have been established for clustering data drawn from disjoint subspaces [30]. However, theoretical analysis in [31] shows that it is possible that SSC over-segments subspaces when the dimensionality of data points is higher than three. Experimental results in [32] show that LRR misclassifies different data points than SSC. Therefore, in order to capture global and the local structure of the data, it is necessary to combine low-rank and sparsity constraints [32,33].

Multi-view subspace clustering can be considered as a part of multi-view or multi-modal learning. Multi-view learning method in [34] learns view generation matrices and representation matrix, relying on the assumption that data from all the views share the same representation matrix. The multi-view method in [35] is based on the canonical correlation analysis in extraction of two-view filter-bank-based features for image classification task. Similarly, in [36] the authors rely on tensor-based canonical correlation analysis to perform multi-view dimensionality reduction. This approach can be used as a preprocessing step in multi-view learning in case of high-dimensional data. In [37] low-rank representation matrix is learned on each view separately and learned representation matrices are concatenated to a matrix from which a unified graph affinity matrix is obtained. The method in [38] relies on learning a linear projection matrix for each view separately. High-order distance-based multi-view stochastic learning is proposed in [39], to efficiently explore the complementary characteristics of multi-view features for image classification. The method in [40] is application oriented towards image reranking and assumes that multi-view features are contained in hypergraph Laplacians that define different modalities. In [41] authors propose multi-view matrix completion algorithm for handling multi-view features in semi-supervised multi-label image classification.

Previous multi-view subspace clustering works [42–45] address the problem by constructing affinity matrix on each view separately and then extend algorithm to handle multi-view data. However, since input data may often be corrupted by noise, this approach can lead to the propagation of noise in the affinity matrices and degrade clustering performance. Different from the existing approaches, we propose multi-view spectral clustering framework that jointly learns a subspace representation by constructing single affinity matrix shared by multi-view data, while at the same time encourages low-rank and sparsity of the representation. We propose Multi-view Low-rank Sparse Subspace Clustering (MLRSSC) algorithms that enforce agreement: (i) between affinity matrices of the pairs of views; (ii) between affinity matrices towards a common centroid. Opposed to [35,40,46], the proposed approach can deal with highly heterogeneous multi-view data coming from different modalities. We present optimization procedure to solve the convex dual optimization problems using Alternating Direction Method of Multipliers (ADMM) [47]. Furthermore, we propose the kernel extension of our algorithms by solving the problem in a Reproducing Kernel Hilbert Space (RKHS). Experimental results show that MLRSSC algorithm outperforms state-of-the-art multi-view subspace clustering algorithms on several benchmark datasets. Additionally, we evaluate performance on a novel real-world heterogeneous multi-view dataset from biological domain.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of the low-rank and sparse subspace clustering methods. Section 3 introduces two novel multi-view subspace clustering algorithms. In Section 4 we present the kernelized version of the proposed algorithms by formulating subspace clustering problem in RKHS. The performance of the new algorithms is demonstrated in Section 5. Section 6 concludes the paper.

Table 1
Notations and abbreviations.

Notation	Definition
N	Number of data points
k	Number of clusters
v	View index
n_v	Number of views
$D^{(v)}$	Dimension of data points in a view v
$\mathbf{X}^{(v)} \in \mathbb{R}^{D^{(v)} \times N}$	Data matrix in a view v
$\mathbf{C}^{(v)} \in \mathbb{R}^{N \times N}$	Representation matrix in a view v
$\mathbf{C}^{*} \in \mathbb{R}^{N \times N}$	Centroid representation matrix
$\mathbf{W} \in \mathbb{R}^{N \times N}$	Affinity matrix
$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$	Singular value decomposition (SVD) of $\mathbf{X}$
$\Phi(\mathbf{X}^{(v)})$	Data points in a view v mapped into high-dimensional feature space
$\mathbf{K}^{(v)} \in \mathbb{R}^{N \times N}$	Gram matrix in a view v

2. Background and related work

In this section, we give a brief introduction to Sparse Subspace Clustering (SSC) [15], Low-Rank Representation (LRR) [16,17] and Low-rank Sparse Subspace Clustering (LRSSC) [32].

2.1. Main notations

Throughout this paper, matrices are represented with bold capital symbols and vectors with bold lower-case symbols. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The ℓ_1 norm, denoted by $\|\cdot\|_1$, is the sum of absolute values of matrix elements; infinity norm $\|\cdot\|_\infty$ is the maximum absolute element value; and the nuclear norm $\|\cdot\|_*$ is the sum of singular values of a matrix. Trace operator of a matrix is denoted by $tr(\cdot)$ and $diag(\cdot)$ is the vector of diagonal elements of a matrix. $\mathbf{0}$ denotes null vector. Table 1 summarizes some notations used throughout the paper.

2.2. Related work

Consider the set of N data points $\mathbf{X} = \left\{\mathbf{x}_i \in \mathbb{R}^D\right\}_{i=1}^N$ that lie in a union of $k > 1$ linear subspaces of unknown dimensions. Given the set of data points $\mathbf{X}$, the task of subspace clustering is to cluster data points according to the subspaces they belong to. The first step is the construction of the affinity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ whose elements define the similarity between data points. Ideally, the affinity matrix is a block diagonal matrix such that a nonzero distance is assigned to the points from the same subspace. LRR, SSC and LRSSC construct the affinity matrix by enforcing low-rank, sparsity and low-rank plus sparsity constraints, respectively.

Low-Rank Representation (LRR) [16,17] seeks to find a low-rank representation matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$ for input data $\mathbf{X}$. The basic model of LRR is the following:

$$\min_{\mathbf{C}} \left\|\mathbf{C}\right\|_* \quad s.t. \quad \mathbf{X} = \mathbf{XC}, \tag{1}$$