Pattern Recognition Letters 43 (2014) 47–61

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Low rank subspace clustering (LRSC)

René Vidal [a,*], Paolo Favaro [b]

[a] Center for Imaging Science, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
[b] Institute of Informatics and Applied Mathematics, University of Bern, 3012, Switzerland

ARTICLE INFO

Article history:
Available online 16 September 2013

Communicated by Kim Boyer

Keywords:
Subspace clustering
Low-rank and sparse methods
Principal component analysis
Motion segmentation
Face clustering

ABSTRACT

We consider the problem of fitting a union of subspaces to a collection of data points drawn from one or more subspaces and corrupted by noise and/or gross errors. We pose this problem as a non-convex optimization problem, where the goal is to decompose the corrupted data matrix as the sum of a clean and self-expressive dictionary plus a matrix of noise and/or gross errors. By self-expressive we mean a dictionary whose atoms can be expressed as linear combinations of themselves with low-rank coefficients. In the case of noisy data, our key contribution is to show that this non-convex matrix decomposition problem can be solved in closed form from the SVD of the noisy data matrix. The solution involves a novel polynomial thresholding operator on the singular values of the data matrix, which requires minimal shrinkage. For one subspace, a particular case of our framework leads to classical PCA, which requires no shrinkage. For multiple subspaces, the low-rank coefficients obtained by our framework can be used to construct a data affinity matrix from which the clustering of the data according to the subspaces can be obtained by spectral clustering. In the case of data corrupted by gross errors, we solve the problem using an alternating minimization approach, which combines our polynomial thresholding operator with the more traditional shrinkage-thresholding operator. Experiments on motion segmentation and face clustering show that our framework performs on par with state-of-the-art techniques at a reduced computational cost.

1. Introduction

The past few decades have seen an explosion in the availability of datasets from multiple modalities. While such datasets are usually very high-dimensional, their intrinsic dimension is often much smaller than the dimension of the ambient space. For instance, the number of pixels in an image can be huge, yet most computer vision models use a few parameters to describe the appearance, geometry and dynamics of a scene. This has motivated the development of a number of techniques for finding low-dimensional representations of high-dimensional data.

One of the most commonly used methods is Principal Component Analysis (PCA), which models the data with *a single* low-dimensional subspace. In practice, however, the data points could be drawn from *multiple subspaces* and the *membership* of the data points to the subspaces could be unknown. For instance, a video sequence could contain several moving objects and different subspaces might be needed to describe the motion of different objects in the scene. Therefore, there is a need to simultaneously cluster the data into multiple subspaces and find a low-dimensional subspace fitting each group of points. This problem, known as *subspace clustering*, finds numerous applications in computer vision, e.g., image segmentation (Yang et al., 2008), motion segmentation (Vidal et al., 2008) and face clustering (Ho et al., 2003), image processing, e.g., image representation and compression (Hong et al., 2006), and systems theory, e.g., hybrid system identification (Vidal et al., 2003b).

1.1. Prior work on subspace clustering

Over the past decade, a number of subspace clustering methods have been developed. This includes algebraic methods (Boult and Brown, 1991; Costeira and Kanade, 1998; Gear, 1998; Vidal et al., 2003a; Vidal et al., 2004; Vidal et al., 2005), iterative methods (Bradley and Mangasarian, 2000; Tseng, 2000; Agarwal and Mustafa, 2004; Lu and Vidal, 2006; Zhang et al., 2009), statistical methods (Tipping and Bishop, 1999,Sugaya and Kanatani, 2004, Gruber and Weiss, 2004,Yang et al., 2006, Ma et al., 2007,Rao et al., 2008, Rao et al., 2010), and spectral clustering-based methods (Boult and Brown, 1991; Yan and Pollefeys, 2006; Zhang et al., 2010; Goh and Vidal, 2007; Elhamifar and Vidal, 2009; Elhamifar and Vidal, 2010; Elhamifar and Vidal, 2013; Liu et al., 2010; Chen and Lerman, 2009). Among them, methods based on spectral clustering have been shown to perform very well for

* Corresponding author. Tel.: +1 410 516 7306; fax: +1 410 516 7775.
E-mail address: rvidal@cis.jhu.edu (R. Vidal).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.08.006

several applications in computer vision (see Vidal (2011) for a review and comparison of existing methods).

Spectral clustering-based methods (see von Luxburg, 2007 for a review) decompose the subspace clustering problem in two steps. In the first step, a symmetric *affinity matrix* $C = [c_{ij}]$ is constructed, where $c_{ij} = c_{ji} \geqslant 0$ measures whether points i and j belong to the same subspace. Ideally $c_{ij} \approx 1$ if points i and j are in the same subspace and $c_{ij} \approx 0$ otherwise. In the second step, a weighted undirected graph is constructed where the data points are the nodes and the affinities c_{ij} are the weights. The segmentation of the data is then found by clustering the eigenvectors of the graph Laplacian using central clustering techniques, such as k-means. Arguably, the most difficult step is to build a good affinity matrix. This is because two points could be very close to each other, but lie in different subspaces (e.g., near the intersection of two subspaces). Conversely, two points could be far from each other, but lie in the same subspace.

Earlier methods for building an affinity matrix (Boult and Brown, 1991; Costeira and Kanade, 1998) compute the singular value decomposition (SVD) of the data matrix $D = U\Sigma V^\top$ and let $C = V_1 V_1^\top$, where the columns of V_1 are the top $r = \text{rank}(D)$ singular vectors of D. The rationale behind this choice is that $c_{ij} = 0$ when points i and j are in different *independent* subspaces and the data are *uncorrupted*, as shown in Vidal et al. (2005). In practice, however, the data are often contaminated by noise and gross errors. In such cases, the equation $c_{ij} = 0$ does not hold, even if the rank of the noiseless D was given. Moreover, selecting a good value for r becomes very difficult, because D is full rank. Furthermore, the equation $c_{ij} = 0$ is derived under the assumption that the subspaces are linear. In practice, many datasets are better modeled by affine subspaces.

More recent methods for building an affinity matrix address these issues by using techniques from sparse and low-rank representation. For instance, it is shown in Elhamifar and Vidal (2009, 2010, 2013) that a point in a union of multiple subspaces admits a sparse representation with respect to the dictionary formed by all other data points, i.e., $D = DC$, where C is sparse. It is also shown in Elhamifar and Vidal (2009, 2010, 2013) that, if the subspaces are independent, the nonzero coefficients in the sparse representation of a point correspond to other points in the same subspace, i.e., if $c_{ij} \neq 0$, then points i and j belong to the same subspace. Moreover, the nonzero coefficients can be obtained by ℓ_1 minimization. These coefficients are then converted into symmetric and nonnegative affinities, from which the segmentation is found using spectral clustering. A very similar approach is presented in Liu et al. (2010). The major difference is that a low-rank representation is used in lieu of the sparsest representation. While the same principle of representing a point as a linear combination of other points has been successfully used when the data are corrupted by noise and gross errors, from a theoretical viewpoint it is not clear that the above methods are effective when using a corrupted dictionary.

1.2. Paper contributions

In this paper, we propose a general optimization framework for solving the subspace clustering problem in the case of data corrupted by noise and/or gross errors. Given a corrupted data matrix $D \in \mathbb{R}^{M\times N}$, we wish to decompose it as the sum of a self-expressive, noise-free and outlier-free (clean) data matrix $A \in \mathbb{R}^{M\times N}$, a noise matrix $G \in \mathbb{R}^{M\times N}$, and a matrix of sparse gross errors $E \in \mathbb{R}^{M\times N}$. We assume that the columns of the matrix $A = [a_1, a_2, \ldots, a_N]$ are points in $\mathbb{R}^M$ drawn from a union of $n \geqslant 1$ low-dimensional linear subspaces of unknown dimensions $\{d_i\}_{i=1}^n$, where $d_i \ll M$. We also assume that A is self-expressive, which means that the clean data points can be expressed as linear combinations of themselves, i.e.,

$$a_j = \sum_{i=1}^{N} a_i c_{ij} \quad \text{or} \quad A = AC, \tag{1}$$

where $C = [c_{ij}]$ is the matrix of coefficients. This constraint aims to capture the fact that a point in a linear subspace can be expressed as a linear combination of other points in the same subspace. Therefore, we expect c_{ij} to be zero if points i and j are in different subspaces.

Notice that the constraint $A = AC$ is non-convex, because both A and C are unknown. This is an important difference with respect to existing methods, which enforce $D = DC$ where D is the dictionary of corrupted data points. Another important difference is that we directly enforce C to be symmetric, while existing methods symmetrize C as a post-processing step.

The proposed framework, which we call Low Rank Subspace Clustering (LRSC), is based on solving the following non-convex optimization problem:

$$(P) \quad \begin{aligned} &\min_{A,C,E,G} \|C\|_* + \tfrac{\tau}{2}\|A - AC\|_F^2 + \tfrac{\alpha}{2}\|G\|_F^2 + \gamma\|E\|_1 \\ &\text{s.t. } D = A + G + E \text{ and } C = C^\top, \end{aligned}$$

where $\|X\|_* = \sum_i \sigma_i(X)$, $\|X\|_F^2 = \sum_{ij} X_{ij}^2$ and $\|X\|_1 = \sum_{ij}|X_{ij}|$ are, respectively, the nuclear, Frobenius and ℓ_1 norms of X. The above formulation encourages:

- C to be low-rank (by minimizing $\|C\|_*$),
- A to be self-expressive (by minimizing $\|A - AC\|_F^2$),
- G to be small (by minimizing $\|G\|_F^2$), and
- E to be sparse (by minimizing $\|E\|_1$).

The main contribution of our work is to show that important particular cases of P (see Table 1) can be solved in closed form from the SVD of the data matrix. In particular, we show that in the absence of gross errors (i.e., $\gamma = \infty$), A and C can be obtained by thresholding the singular values of D and A, respectively. The thresholding is done using a novel polynomial thresholding operator, which reduces the amount of shrinkage with respect to existing methods. Indeed, when the self-similarity constraint $A = AC$ is enforced exactly (i.e., $\alpha = \infty$), the optimal solution for A reduces to classical PCA, which does not perform any shrinkage. Moreover, the optimal solution for C reduces to the affinity matrix for subspace clustering proposed by Costeira and Kanade (1998). In the case of data corrupted by gross errors, a closed-form solution appears elusive. We thus use an augmented Lagrange multipliers method. Each iteration of our method involves a polynomial thresholding of the singular values to reduce the rank and a regular shrinkage-thresholding to reduce gross errors.

Table 1
Particular cases of P solved in this paper.

Relaxed	Exact
Uncorrupted	
P_1: Section 3.1	P_2: Section 3.2
$0 < \tau < \infty$	$\tau = \infty$
$\alpha = \infty$	$\alpha = \infty$
$\gamma = \infty$	$\gamma = \infty$
Noise	
P_3: Section 4.1	P_4: Section 4.2
$0 < \tau < \infty$	$\tau = \infty$
$0 < \alpha < \infty$	$0 < \alpha < \infty$
$\gamma = \infty$	$\gamma = \infty$
Gross errors	
P_5: Section 5.1	P_6: Section 5.2
$0 < \tau < \infty$	$\tau = \infty$
$0 < \alpha < \infty$	$0 < \alpha < \infty$
$0 < \gamma < \infty$	$0 < \gamma < \infty$