Sparse Representation for Manifold Clustering and Dimensionality Reduction

THESIS PROPOSAL

Ehsan Elhamifar
Department of Electrical and Computer Engineering
Johns Hopkins University

Committee:

Dr. René Vidal
Dr. Sanjeev Khudanpur
Dr. Trac Tran

1 Introduction

In many areas of machine learning, computer vision, image processing, data mining, and information retrieval one is confronted with intrinsically low-dimensional data on multiple manifolds embedded in a very high dimensional space. In computer vision, for example, the feature point trajectories associated with a moving object in a video sequence lie in a manifold of very low dimension. Thus, the collection of feature trajectories associated with multiple moving objects lie in a union of low-dimensional manifolds.

Two fundamental tasks associated with modeling high-dimensional data lying on multiple low-dimensional manifolds are dimensionality reduction and clustering.

Dimensionality reduction. This task is concerned with finding a meaningful low-dimensional representation of the high-dimensional observations. Examples include finding a meaningful low-dimensional representation for images of a person performing multiple gestures such as laughing, frowning, turning right or left, etc, or finding a meaningful representation for the words in a document such that the words which share common features are also close in the low-dimensional representation. Dimensionality reduction methods can be divided in two groups: linear and nonlinear methods. Linear dimensionality reduction algorithms such as principal component analysis (PCA) [17] and multi-dimensional scaling (MDS) [7], reduce the dimension of the data by modeling it with a low-dimensional affine subspace. Nonlinear dimensionality reduction algorithms such as ISOMAP [24], Locally Linear Embedding (LLE) [22], and Laplacian Eigenmaps [2], reduce the dimensionality of the data by modeling it with a nonlinear manifold. However, there are a number of challenges associated with the state-of-the-art dimensionality reduction techniques. (1) Methods such as PCA and MDS only deal with data lying on flat manifolds. Thus, their performance drastically deteriorates when the data lie in highly nonlinear manifolds. (2) ISOMAP, computes a similarity matrix by finding the geodesic distances between pairs of points on the manifold which is costly. In addition, it computes the low-dimensional embedding by finding the eigenvectors of a dense similarity matrix, which is also a costly operation. (3) LLE and Laplacian Eigenmaps choose only the K nearest neighbors of each data point on the manifold, which results in a sparse matrix from which the low-dimensional embedding can be computed. However, choosing the right number of neighbors is a challenging problem that affects the quality of the embedding.

Manifold clustering. In this task, we are given data lying in multiple manifolds and the goal is to segment the data points so that points in the same manifold belong to the same group. For instance, in the example of multiple rigid-body motions in a video, we want to cluster the feature trajectories of points on multiple moving objects, so that the points in the same moving object belong to the same group. Other examples include clustering face images of different people, temporal segmentation of videos, document clustering, etc. A special but important class of manifolds are affine subspaces, which are flat manifolds. In fact, several of the tasks mentioned above can be cast as clustering of data on multiple affine subspaces. Most existing manifold clustering methods such as GPCA [27], SCC [5], LSA [29], MPPCA [26], and RANSAC [14] address this special case. LLMC [15] is one of the few existing methods that addresses the case of data lying in multiple nonlinear manifolds. Unfortunately, there are a number of issues associated with the state-of-the-art manifold clustering algorithms. (1) The computational complexity of methods such as GPCA, SCC increases exponentially by increasing the number of subspaces and/or their dimensions. (2) Methods such as LLMC and LSA have difficulties dealing with intersecting manifolds. (3) Methods such as MPPCA and RANSAC assume the dimension of the subspaces to be known, while in many

tasks we do not have this information beforehand. (4) The performance of most existing methods decreases drastically when the data are contaminated with noise and outliers, which is an important problem arising in many applications.

The goal of this thesis is to develop a mathematical framework for simultaneous dimensionality reduction and manifold clustering that addresses these challenges. More specifically, we propose a computationally efficient algorithm for clustering intersecting and non-intersecting manifolds contaminated with noise and outliers. The method we propose reduces the dimensionality of the data by choosing a few neighbors of each data point on a manifold. However, unlike the state-of-the-art dimensionality reduction algorithms, the proposed mechanism automatically chooses the neighbors of each data point without specifying *a priori* the number of neighbors. Moreover, the neighbors are automatically chosen as points in the same manifold rather than using a distance in the ambient space.The proposed framework relies on recent advances in sparse representation theory, which addresses the problem of recovering a sparse representation of signals in an appropriate basis from a limited number of measurements. This finds numerous applications in signal and image processing and machine learning, such as signal/image/video compression, learning a dictionary for a collection of images/textures, sparse regression, etc. The main focus of the research in this area has been to find computationally efficient algorithms for extracting a sparse representation of signals and vectors together with the conditions under which the methods are guaranteed to recover the sparse representation with high probability. The state-of-the-art sparse recovery methods address the problem of sparse representation of signals in a single basis or in multiple known subspaces. Our goal is to extend these results in several dimensions. First, we plan to find efficient methods for finding the sparse representation of data lying on multiple subspaces without knowing the basis of each subspace. Second, we plan to study conditions on subspace arrangements and the distribution of the data under which our proposed sparse recovery method is guaranteed to recover the true sparse representation of the data points. Third, we plan to address the general case of clustering data lying in multiple nonlinear manifolds as well as reducing the dimensionality of the data on each manifold using the sparse recovery methods. Fourth, we plan to evaluate the efficacy of the proposed algorithm for several applications in computer vision and image processing.

In what follows, we shall review in Section 2, the state of the art techniques for manifold clustering as well as sparse recovery methods. In Section 3, we present our proposed method for subspace clustering which is based on sparse representation techniques together with experimental results on the motion segmentation problem. In Section 4, we discuss our future research plan which covers a range of theoretical problems and applications.

2 State of the Art

Manifold clustering is an important problem with numerous applications in image processing, *e.g.*image representation and compression [16, 30], and computer vision, *e.g.*image/motion/video segmentation [6, 18, 31, 29, 28]. In what follows, we review the state of the art on manifold clustering. Then, we briefly review the sparse recovery methods which also play a central role in our research for attacking the manifold clustering problem.

2.1 Subspace clustering

A special but important class of manifolds are flat manifolds, also called affine subspaces. Given a set of points drawn from a union of subspaces, the goal of subspace clustering is to find the number of subspaces, their dimensions, a basis for each subspace, and the segmentation of the data. State-of-the-art subspace clustering methods can be divided into four main categories: algebraic, spectral clustering, statistical, and compression-based methods.

Algebraic Methods. Generalized Principal Component Analysis (GPCA) [27] is an algebraic method for clustering data lying in multiple subspaces. The main idea behind GPCA is that one can fit a union of n subspaces with a set of polynomials of degree n, whose derivatives at a point give a vector normal to the subspace containing that point. The segmentation of the data is then obtained by grouping these normal vectors, which can be done using several techniques. In theory, GPCA can deal with subspaces in any relative configuration and does not require the number of subspaces or their dimensions to be known beforehand. In practice, however, GPCA is sensitive to noise and outliers, and its complexity increases exponentially with the number of subspaces and their dimensions.

Spectral Clustering Methods. Spectral-clustering based methods use local information around each point to build a similarity matrix between pairs of points. The segmentation of the data is then obtained by applying spectral clustering to this similarity matrix. In Locally Linear Manifold Clustering (LLMC) [15], the similarity is built from the coefficients obtained after writing each point as a linear combination of its nearest neighbors. In Local Subspace Affinity (LSA) [29] the similarity is built by fitting an affine subspace to each point and its nearest neighbors, and then computing the angles between these locally estimated subspaces. Both methods have difficulties dealing with points near the intersection of two subspaces, because the neighborhood of a point can contain points from different subspaces. Spectral Curvature Clustering (SCC) [5] resolves this issue by looking at multi-way similarities that capture the curvature of a collection of points within an affine subspace. However, the complexity of building a multi-way similarity grows exponentially with the number of subspaces and their dimensions. Another disadvantage of most spectral-clustering based methods is that they require the number of subspaces to be known and the subspace dimensions to be known and equal.

Statistical Methods. Statistical approaches, such as Mixtures of Probabilistic PCA (MPPCA) [26] and Multi-Stage Learning (MSL) [23], assume that the distribution of the data inside each subspace is degenerate Gaussian and alternate between data clustering and subspace estimation by applying Expectation Maximization (EM) to a mixture of probabilistic PCAs. The main drawbacks of both approaches are that they generally require the number and dimensions of the subspaces to be known, and that they are sensitive to correct initialization. Robust methods, such as RANdom SAmple Consensus (RANSAC) [14], fit a subspace of dimension d to randomly chosen subsets of d points until the number of inliers is large enough. The inliers are then removed and the process is repeated to find a second subspace, and so on. RANSAC can deal with noise and outliers, and does not need to know the number of subspaces. However, the dimensions of the subspaces must be known and equal, and the number of trials needed to find d points in the same subspace grows exponentially with the number and dimension of the subspaces.

Compression-Based Methods. Compression-based approaches, such as Agglomerative Lossy Compression (ALC) [20], model each subspace with a degenerate Gaussian and look for the segmentation of the data that minimizes the coding length needed to fit these points with a mixture

of Gaussians. As this minimization problem is NP hard, a suboptimal solution is found by first assuming that each point forms its own group, and then iterative merging pairs of groups to reduce the coding length. ALC can handle noise and outliers in the data, and can estimate the number of subspaces and their dimensions. However, there is no theoretical proof for the optimality of the algorithm.

2.2 Sparse representation and compressed sensing

Compressed sensing (CS) is based on the idea that many signals or vectors can have a concise representation when expressed in a proper basis. As a result, the information rate of a sparse signal can be much smaller than the rate suggested by its maximum frequency. In this section, we review recently developed techniques from CS for sparsely representing signals lying in one or more subspaces.

Sparse representation in a single subspace. Consider a system of linear equations of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{s}$ where $\boldsymbol{A} \in \mathbb{R}^{D\times N}$, called the measurement matrix, has more columns than rows, *i.e.*, $D < N$. This underdetermined system of linear equations has many solutions for a given vector $\boldsymbol{y} \in \mathbb{R}^D$. However, there are cases where we are interested in finding the sparsest solution. In principle, such a sparse representation can be obtained by solving the optimization problem:

$$\min \|\boldsymbol{s}\|_0 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{s}, \tag{1}$$

where $\|\boldsymbol{s}\|_0$ is the ℓ_0 norm of $\boldsymbol{s}$, *i.e.*, the number of nonzero elements in $\boldsymbol{s}$. However, such an optimization problem is in general non-convex and NP-hard. This has motivated the development of several methods for efficiently extracting a sparse representation of signals/vectors. One of the well-known methods is the Basis Pursuit (BP) algorithm, which replaces the non-convex optimization in (1) by the following convex ℓ_1 optimization problem [8]:

$$\min \|\boldsymbol{s}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{s}. \tag{2}$$

The works of [4, 3] show that we can recover perfectly a sparse signal/vector by using the BP algorithm in (2) under certain conditions on the so-called *isometry constant* of the $\boldsymbol{A}$ matrix.

Sparse representation in a union of subspaces. Most of the work on CS deals with sparse representation of signals/vectors lying in a single linear subspace. The more general case where the signals/vectors lie in a union of low-dimensional linear subspaces was only recently considered. The work of Eldar [11] shows that when the subspaces are disjoint (intersect only at the origin), a basis for each subspace is known, and certain condition on a modified isometry constant holds, one can recover the block-sparse vector $\boldsymbol{s}$ exactly by solving an ℓ_1/ℓ_2 optimization problem.

More precisely, let $\{\boldsymbol{A}_i \in \mathbb{R}^{D\times d_i}\}_{i=1}^n$ be a set of bases for n disjoint linear subspaces embedded in $\mathbb{R}^D$ with dimensions $\{d_i\}_{i=1}^n$. If $\boldsymbol{y}$ belongs to the i-th subspace, we can represent it as the sparse solution of

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{s} = [\boldsymbol{A}_1,\ \boldsymbol{A}_2, \cdots,\ \boldsymbol{A}_n]\,[\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \cdots, \boldsymbol{s}_n^\top]^\top, \tag{3}$$

where $\boldsymbol{s}_i \in \mathbb{R}^{d_i}$ is a nonzero vector and all other vectors $\{\boldsymbol{s}_j \in \mathbb{R}^{d_j}\}_{j\neq i}$ are zero. Therefore, $\boldsymbol{s}$ is the solution to the following non-convex optimization problem:

$$\min \sum_{i=1}^{n} 1(\|\boldsymbol{s}_i\|_2 > 0) \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{s}, \tag{4}$$

where $1(\|\boldsymbol{s}_i\|_2 > 0)$ is an indicator function that takes the value 1 when $\|\boldsymbol{s}_i\|_2 > 0$ and zero otherwise. [11] shows that if a modified isometry constant of the matrix $\boldsymbol{A}$ satisfies a certain condition, then the solution to the (convex) ℓ_2/ℓ_1 program

$$\min \sum_{i=1}^{n} \|\boldsymbol{s}_i\|_2 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{s} \tag{5}$$

coincides with that of (4).

3 Preliminary Results

Our recent work on sparse subspace clustering (SSC) [12, 13] proposes a new subspace clustering method based on sparse representation. Unlike the conventional sparse recovery methods, the data do not have a sparse representation in a single basis but in a union of bases. Also, in contrast to the block-sparse recovery methods where the given points have sparse representations in a union of *known bases*, we do not know the bases for subspaces nor do we know which data belongs to which subspace.

We tackle the subspace clustering problem by introducing a method which is based on sparse recovery. The proposed method is based on the observation that each data point on a subspace can write itself as a linear combination of all other data points. However, generically, the sparsest representation comes from points in the same subspace. We show that, under certain conditions on the arrangement of subspaces and the distribution of the data within each subspace, our method is guaranteed to recover a sparse representation of a point as a linear combination of points from the same subspace. We use this sparse representation to build a similarity graph from which we can infer the segmentation of the data.

3.1 Clustering linear subspaces

Let $\{S_i\}_{i=1}^n$ be an arrangement of n linear subspaces of $\mathbb{R}^D$ of dimensions $\{d_i\}_{i=1}^n$. Consider now a given collection of $N = \sum_{i=1}^n N_i$ noise-free data points, $\{\boldsymbol{y}_i\}_{i=1}^N$, drawn from the n subspaces $\{S_i\}_{i=1}^n$. We denote the matrix whose columns are the N_i points drawn from subspace S_i as $\boldsymbol{Y}_i \in \mathbb{R}^{D\times N_i}$ and the matrix containing all the data points as $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{Y}_1, & \ldots, & \boldsymbol{Y}_n\end{bmatrix}\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times N}$ is an unknown permutation matrix. We assume that we do not know *a priori* the bases for each one of the subspaces nor do we know which data points belong to which subspace. We are interested in finding the number of subspaces, their dimensions, a basis for each subspace, and the segmentation of the data from $\boldsymbol{Y}$.

The SSC algorithm (see [12, 13]) is based on the observation that each data point $\boldsymbol{y}_i \in S_j$ can always be written as a linear combination of all the other data points in $\{S_i\}_{i=1}^n$. However, generically, the *sparsest* representation is obtained when the point $\boldsymbol{y}_i$ is written as a linear combination of points in its own subspace. Finding such a sparse representation is an NP hard problem since it requires solving the following non-convex optimization problem

$$\min \|\boldsymbol{c}_i\|_0 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i, \tag{6}$$

where $\boldsymbol{Y}_{\widehat{i}} \in \mathbb{R}^{D\times N-1}$ denotes the matrix obtained from $\boldsymbol{Y}$ by removing its i-th column, $\boldsymbol{y}_i$. In [12, 13], we show that under certain conditions on the arrangement of subspaces and the distribution

of the data within each subspace, we can find the sparse representation efficiently using the following ℓ_1 optimization problem

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i. \tag{7}$$

Theorem 1 *Let $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$ be a matrix whose columns are drawn from a union of n independent [1] linear subspaces. Assume that the points within each subspace are in general position. Let $\boldsymbol{y}_i$ be a data point in subspace S_k. The solution to the ℓ_1 problem in* (7)*, $\boldsymbol{c}_i$, is sparse and its non-zero entries c_{ij} correspond to data points $\boldsymbol{y}_j$ in subspace S_k.*

Requiring the subspaces to be independent might be a strong assumption in some applications. For instance, when segmenting multiple rigid-body motions in a video sequence, the subspaces become partially dependent for articulated objects, or for objects moving in a common plane, as shown in [28]. In [13] we relax the independence assumption and consider the more general class of disjoint subspaces (each pair of subspaces only intersect at the origin). We derive a condition on the subspace angles and the distribution of the data across subspaces under which SSC is guaranteed to recover the sparse representation of each data point as a linear combination of other data points in the same subspace.

The optimal solution, $\boldsymbol{c}_i \in \mathbb{R}^{N-1}$, in equation (7) is a vector whose nonzero entries correspond to points (columns) in $\boldsymbol{Y}_{\widehat{i}}$ that lie in the same subspace as $\boldsymbol{y}_i$. Thus, by inserting a zero entry at the i-th row of $\boldsymbol{c}_i$, we make it an N-dimensional vector, $\widehat{\boldsymbol{c}}_i \in \mathbb{R}^N$, whose nonzero entries correspond to points in $\boldsymbol{Y}$ that lie in the same subspace as $\boldsymbol{y}_i$.

After solving (7) at each point $i \in \{1, \ldots, N\}$, we obtain a matrix of coefficients $\boldsymbol{C} = [\widehat{\boldsymbol{c}}_1, \cdots, \widehat{\boldsymbol{c}}_N] \in \mathbb{R}^{N\times N}$. We use this matrix to define a directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. The vertices of the graph $\boldsymbol{V}$ are the N data points and there is an edge $(v_i, v_j) \in \boldsymbol{E}$ when the data point $\boldsymbol{y}_j$ is one of the vectors in the sparse representation of $\boldsymbol{y}_i$, *i.e.*when $C_{ji} \neq 0$. One can easily see that $\boldsymbol{C}$ is the adjacency matrix of $\boldsymbol{G}$.

In general, $\boldsymbol{G}$ is an unbalanced digraph. To make it balanced, we build a new graph $\widetilde{\boldsymbol{G}}$ with the adjacency matrix $\widetilde{\boldsymbol{C}}$, where $\widetilde{C}_{ij} = |C_{ij}| + |C_{ji}|$. $\widetilde{\boldsymbol{C}}$ is still a valid representation of the similarity, because if $\boldsymbol{y}_i$ can write itself as a linear combination of some points including $\boldsymbol{y}_j$ (all in the same subspace), then $\boldsymbol{y}_j$ can also write itself as a linear combination of some points in the same subspace including $\boldsymbol{y}_i$.

Having formed the similarity graph $\widetilde{\boldsymbol{G}}$, we expect that all vertices representing the data points in the same subspace form a connected component in the graph, while the vertices representing points in different subspaces have no edges between them. Therefore, in the case of n subspaces, $\widetilde{\boldsymbol{C}}$ has the following block diagonal form

$$\widetilde{\boldsymbol{C}} = \begin{bmatrix} \widetilde{\boldsymbol{C}}_1 & 0 & \cdots & 0 \\ 0 & \widetilde{\boldsymbol{C}}_2 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & \widetilde{\boldsymbol{C}}_n \end{bmatrix} \boldsymbol{\Gamma}, \tag{8}$$

where $\boldsymbol{\Gamma}$ is a permutation matrix. The Laplacian matrix of $\widetilde{\boldsymbol{G}}$ is then formed as $\boldsymbol{L} = \boldsymbol{D} - \widetilde{\boldsymbol{C}}$, where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $D_{ii} = \sum_j \widetilde{C}_{ij}$.

[1] A collection of n linear subspaces $\{S_i \subset \mathbb{R}^D\}_{i=1}^n$ are independent if $\dim(\bigoplus_{i=1}^n S_i) = \sum_{i=1}^n \dim(S_i)$, where $\oplus$ is the direct sum.

Using the results from spectral graph theory, if the graph has n connected components, the components can be determined from the eigenspace of the zero eigenvalues by applying K-means to the n eigenvectors of the Laplacian corresponding to the smallest eigenvalues.

In summary, the SSC algorithm for linear subspaces proceeds as follows.

Algorithm 1 : Sparse Subspace Clustering (SSC)

Input: A set of points $\{\boldsymbol{y}_i\}_{i=1}^N$ lying in n subspaces $\{S_i\}_{i=1}^n$.

1: For every data point $\boldsymbol{y}_i$, solve the following optimization problem:

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i$$

where $\boldsymbol{Y}_{\widehat{i}} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i+1}, \ldots, \boldsymbol{y}_N]$.

2: Form a similarity graph with N nodes representing the N data points. Connect node i, representing $\boldsymbol{y}_i$, to the other $N-1$ nodes by edge weights $\widetilde{C}_{ij} = |C_{ij}| + |C_{ji}|$.

3: Form the graph Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$. Infer the segmentation of the data from the n eigenvectors of $\boldsymbol{L}$ corresponding to the n smallest eigenvalues using the K-means algorithm [10].

Output: Segmentation of the data: $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n$.

3.2 Clustering affine subspaces

In many cases we need to cluster data lying in multiple affine rather than linear subspaces. For instance, the motion segmentation problem (to be discussed in the next section) involves clustering of data lying on multiple 3-dimensional affine subspaces. However, most existing motion segmentation algorithms deal with this problem by clustering the data as if they belonged to multiple 4-dimensional linear subspaces.

Notice that in the case of affine subspaces, a point can no longer write itself as a linear combination of points in the same subspace. However, we can still write a point $\boldsymbol{y}_i$ as an affine combination of other points, *i.e.*,

$$\boldsymbol{y}_i = \sum_{j \neq i} c_j \boldsymbol{y}_j, \quad \sum_{j \neq i} c_j = 1. \tag{9}$$

In [12], we show that one can recover the sparse representation of data points on affine subspaces by using the following modified BP algorithm for each data point $\boldsymbol{y}_i$,

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i \ \text{ and } \ \boldsymbol{c}_i^\top \mathbf{1} = 1, \tag{10}$$

and form the graph $\widetilde{\boldsymbol{G}}$ from the sparse coefficients. We then apply spectral clustering to the corresponding Laplacian matrix in order to get the segmentation of data.

3.3 Application to motion segmentation

We apply SSC to the motion segmentation problem, *i.e.*, the problem of separating a video sequence into multiple spatiotemporal regions corresponding to different rigid-body motions in the scene. This problem is often solved by extracting a set of points in an image, and tracking these points through the video. Under the affine projection model, all the trajectories associated with a single

Table 1: Classification errors (%) for sequences with 2 motions

	GPCA	LLMC	LSA	SCC	RANSAC	MSL	ALC	SSC-B	SSC-N
Checkerboard	6.09	3.96	2.57	1.30	6.52	4.46	1.55	**0.83**	1.12
Traffic	1.41	3.53	5.43	1.07	2.55	2.23	1.59	0.23	**0.02**
Articulated	2.88	6.48	4.10	3.68	7.25	7.23	10.70	1.63	**0.62**
All	4.59	4.08	3.45	1.46	5.56	4.14	2.40	**0.75**	0.82

Table 2: Classification errors (%) for sequences with 3 motions

	GPCA	LLMC	LSA	SCC	RANSAC	MSL	ALC	SSC-B	SSC-N
Checkerboard	31.95	8.48	5.80	5.68	25.78	10.38	5.20	4.49	**2.97**
Traffic	19.83	6.04	25.07	2.35	12.83	1.80	7.75	0.61	**0.58**
Articulated	16.85	9.38	7.25	10.94	21.38	2.71	21.08	1.60	**1.42**
All	28.66	8.04	9.73	5.31	22.94	8.23	6.69	3.55	**2.45**

rigid motion live in a 3-dimensional affine subspace. Therefore, the motion segmentation problem reduces to clustering a collection of point trajectories according to multiple affine subspaces.

We evaluate SSC on the Hopkins155 motion database, which is available online at `http://www.vision.jhu.edu/data/hopkins155`. The database consists of 155 sequences of two and three motions which can be divided into three main categories: checkerboard, traffic, and articulated sequences. A customary preprocessing step used by other motion segmentation algorithms is to reduce the dimension of the data. As we want a projection that preserves the sparsity of the data, we use a random projection matrix whose entries come from a Normal or a Bernoulli distribution with certain parameters [9, 1]. Table 1 and 2 show our results for sequences having two and three motions, respectively. SSC-B and SSC-N indicate our results for random projections corresponding to the Bernoulli and Gaussian distributions, respectively. Clearly, our method outperforms state-of-the-art methods such as GPCA [27], LLMC [15], LSA [29], SCC [5], RANSAC [14], MSL [23], and ALC [21]. Overall, the SSC algorithm achieves a misclassification rate of 1.24% for the whole database, which is a significant improvement over the state of the art. Figure 1 shows the adjacency matrix and the similarity graph for a video sequence (cars9) in the database. notice that SSC has successfully recovered the sparse representation of the data points, since almost all nonzero coefficients belong to the true subspace. Also the data points in the same subspace form a connected component in the similarity graph. A few number of edges exist between different groups, which are ignored after applying the spectral clustering.

4 Proposed Research

In the previous section, we proposed an algorithm for clustering data lying in a union of subspaces. We showed that for the motion segmentation problem, the algorithm significantly outperforms state-of-the-art subspace clustering methods. Our ongoing research covers a range of theoretical problems as well as different applications.

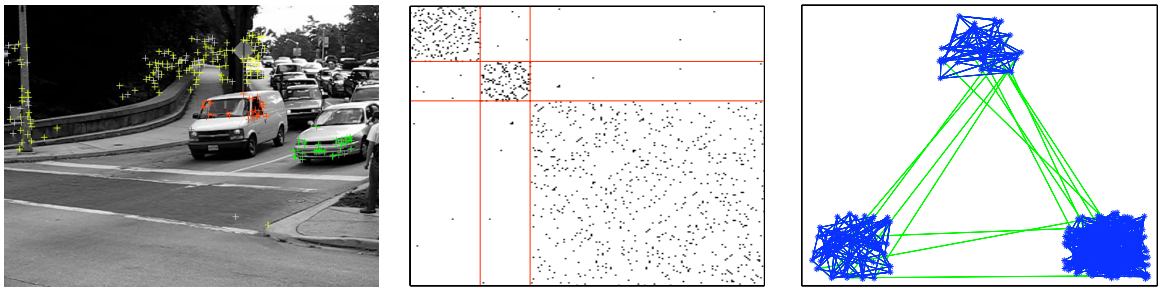

Figure 1: Sparse coefficients and similarity graph obtained by SSC for a video (cars9) in the Hopkins155 database.

4.1 Subspace clustering with noisy data

Consider the case where the data points drawn from a collection of linear or affine subspaces are contaminated with noise. More specifically, let $\bar{\boldsymbol{y}}_i = \boldsymbol{y}_i + \boldsymbol{\zeta}_i$ be the i-th data point corrupted with noise $\boldsymbol{\zeta}_i$ bounded by $\|\boldsymbol{\zeta}_i\|_2 \leq \epsilon$. In order to recover the sparse representation of $\bar{\boldsymbol{y}}_i$, we can look for the sparsest solution of $\bar{\boldsymbol{y}}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i$ with an error of at most ϵ, *i.e.*$\|\boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i - \bar{\boldsymbol{y}}_i\|_2 \leq \epsilon$. In fact, we are interested to solve

$$\min \|\boldsymbol{c}_i\|_0 \quad \text{subject to} \quad \|\boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i - \bar{\boldsymbol{y}}_i\|_2 \leq \epsilon. \tag{11}$$

In the conventional sparse representation theory [3], it has been shown that for the case of noisy data, under certain conditions, one can recover the sparse solution of an underdetermined system of equations by substituting the ℓ_0 with the ℓ_1 norm. For the problem of subspace clustering with noisy data, we propose to find the sparse representation by solving the following convex optimization program

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{subject to} \quad \|\boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i - \bar{\boldsymbol{y}}_i\|_2 \leq \epsilon. \tag{12}$$

However, in many situations we do not know the noise level ϵ beforehand. In such cases we can use the Lasso optimization algorithm [25] to recover the sparse solution from

$$\min \|\boldsymbol{c}_i\|_1 + \gamma \|\boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i - \bar{\boldsymbol{y}}_i\|_2 \tag{13}$$

where $\gamma > 0$ is a constant. In [13] we have shown that for a collection of noise-free data points lying in a union of disjoint subspaces, under appropriate conditions on subspace angles and the distribution of the data across subspaces, we can recover the sparse representation of a point as a combination of points in the same subspace. In our future research, we shall consider the case of noisy data and investigate conditions under which the nonzero coefficients of the sparse representation of a noisy data point come from points in the same subspace *i.e.*, we investigate conditions for the equivalence of (11) and (12). Segmentation of the data into different subspaces follows by applying spectral clustering to the similarity graph formed by the sparse coefficients.

4.2 Subspace clustering with missing data

In practice, some of the entries of the data points may be missing (incomplete data). In motion segmentation, for example, due to occlusions or limitations of the tracker, we may loose some feature points in some of the frames (missing entries). As suggested in [21], we can fill in missing entries using sparse representation techniques. We suggest that one can also cluster data points with missing entries using a sparse representation.

Let $I_i \subset \{1, \ldots, D\}$ denote the indices of missing entries in $\boldsymbol{y}_i \in \mathbb{R}^D$. Let $\boldsymbol{Y}_{\widehat{i}} \in \mathbb{R}^{D \times N-1}$ be obtained by eliminating the vector $\boldsymbol{y}_i$ from the i-th column of the data matrix $\boldsymbol{Y}$. We then form

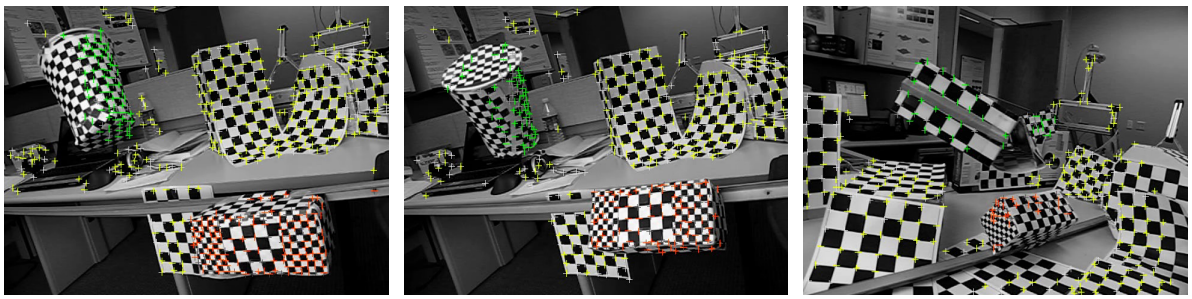

Figure 2: Example frames from three video sequences with incomplete or corrupted trajectories.

$\widetilde{\boldsymbol{y}}_i \in \mathbb{R}^{D-|I_i|}$ and $\widetilde{\boldsymbol{Y}}_{\widehat{i}} \in \mathbb{R}^{D-|I_i| \times N-1}$ by eliminating rows of $\boldsymbol{y}_i$ and $\boldsymbol{Y}_{\widehat{i}}$ indexed by I_i, respectively. Assuming that $\widetilde{\boldsymbol{Y}}_{\widehat{i}}$ is complete, we can find a sparse representation, $\boldsymbol{c}_i^*$, for $\widetilde{\boldsymbol{y}}_i$ by solving the following optimization program

$$\min \|\boldsymbol{c}_i\|_0 \quad \text{subject to} \quad \widetilde{\boldsymbol{y}}_i = \widetilde{\boldsymbol{Y}}_{\widehat{i}} \boldsymbol{c}_i. \tag{14}$$

Note that after eliminating the elements indexed by I_i from the data, we obtain new data points which no longer live in the original subspaces. Part of the future research is to investigate conditions under which we can recover the sparse solution of the non-convex optimization problem in (14) using the following convex relaxation

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{subject to} \quad \widetilde{\boldsymbol{y}}_i = \widetilde{\boldsymbol{Y}}_{\widehat{i}} \boldsymbol{c}_i. \tag{15}$$

The missing entries of $\boldsymbol{y}_i$ are then given by $\boldsymbol{y}_i^* = \boldsymbol{Y}_{\widehat{i}} \boldsymbol{c}_i^*$. Notice that this method for completion of missing data is essentially the same as our method for computing the sparse representation from complete data in (7). Hence we can use the sparse coefficient vectors $\{\boldsymbol{c}_i^*\}_{i=1}^N$ to build the similarity graph and find the segmentation of the data.

We have examined the robustness of SSC to missing data. We use twelve sequences from [28], with nine sequences of two motions and three sequences of three motions, as shown in Figure 2. We use the data points in the original ambient space without projecting them into lower dimensions. For incomplete trajectories, we apply SSC to video sequences between 4% and 35% of whose entries are missing. We compare SSC with Power Factorization-based ALC and ℓ_1-based ALC [21] in Table 3. Our method achieves a misclassification error of 0.13%, which is a significant improvement to the state of the art. Part of the ongoing research is to evaluate the performance of the method for clustering of data with missing entries on a larger database.

Table 3: Misclassifications rates comparison on 12 real motion sequences with missing data.

Method	PF+ ALC_5	PF+ALC_{sp}	ℓ^1+ALC_5	ℓ^1+ALC_{sp}	SSC-N
Average	1.89%	10.81%	3.81%	1.28%	**0.13%**
Median	0.39%	7.85%	0.17%	1.07%	0.00%

4.3 Subspace clustering with corrupted data

In real applications, some of the entries of the data points may be corrupted (outliers). For example, in motion segmentation, due to occlusions or limitations of the tracker, the tracker may loose track

of some features, leading to gross errors. Assume that a few entries of each data point are corrupted. We can also use the sparse representation to correct such entries. More precisely, let $\widetilde{\boldsymbol{y}}_i \in \mathbb{R}^D$ be a corrupted vector obtained from $\bar{\boldsymbol{y}}_i = \boldsymbol{y}_i + \boldsymbol{\zeta}_i$ by adding a sparse error vector $\boldsymbol{e}_i \in \mathbb{R}^D$ as $\widetilde{\boldsymbol{y}}_i = \boldsymbol{y}_i + \boldsymbol{\zeta}_i + \boldsymbol{e}_i$. We can then write

$$\widetilde{\boldsymbol{y}}_i = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i + \boldsymbol{e}_i = [\boldsymbol{Y}_{\widehat{i}} \;\; \boldsymbol{I}_D] \begin{bmatrix} \boldsymbol{c}_i \\ \boldsymbol{e}_i \end{bmatrix} + \boldsymbol{\zeta}_i,$$

where the coefficient vector $[\boldsymbol{c}_i^\top, \boldsymbol{e}_i^\top]^\top$ is still sparse, and hence can be recovered from

$$\min \left\| \begin{bmatrix} \boldsymbol{c}_i \\ \boldsymbol{e}_i \end{bmatrix} \right\|_0 \quad \text{subject to} \quad \left\| \widetilde{\boldsymbol{y}}_i - [\boldsymbol{Y}_{\widehat{i}} \;\; \boldsymbol{I}_D] \begin{bmatrix} \boldsymbol{c}_i \\ \boldsymbol{e}_i \end{bmatrix} \right\|_2 < \epsilon. \tag{16}$$

We investigate conditions under which we can recover the true sparse representation from the following convex program

$$\min \left\| \begin{bmatrix} \boldsymbol{c}_i \\ \boldsymbol{e}_i \end{bmatrix} \right\|_1 \quad \text{subject to} \quad \left\| \widetilde{\boldsymbol{y}}_i - [\boldsymbol{Y}_{\widehat{i}} \;\; \boldsymbol{I}_D] \begin{bmatrix} \boldsymbol{c}_i \\ \boldsymbol{e}_i \end{bmatrix} \right\|_2 < \epsilon. \tag{17}$$

Also, we investigate the extent up to which we can have outlying entries in the data and still can recover the true sparse solution. We can then recover the original vector without outliers as $\boldsymbol{y}_i^* = \boldsymbol{Y}_{\widehat{i}}\boldsymbol{c}_i^*$. As before, the segmentation of the data would be obtained from the sparse coefficients $\{\boldsymbol{c}_i^*\}_{i=1}^N$ using spectral clustering.

For corrupted trajectories, we apply SSC to the sequences between 4% and 35% of whose entries are corrupted. Our results in Table 4 compared with the results of ℓ_1-based ALC indicate the robustness of SSC to outliers. In contrast to ALC, we do not need to use l_1 as an initialization step to complete the trajectories and then apply the segmentation algorithm. The resulting sparse coefficients are used directly to build the similarity graph and do the spectral clustering. Part of the ongoing research is to evaluate the robustness of the method to outliers on a larger database .

Table 4: Misclassifications rates comparison on 12 real motion sequences with corrupted trajectories.

Method	$\ell^1 + \text{ALC}_5$	$\ell^1 + \text{ALC}_{\text{sp}}$	SSC-N
Average	4.15%	3.02%	**1.05%**
Median	0.21%	0.89%	0.43%

Temporal video segmentation. Some video sequences may consist of several scenes or activities which are with a reasonable extent disjoint from each other. A common assumption in such cases is that if we treat each image frame as a data point (for example by vectorizing each 2D image), the images corresponding to the same scene or activity would span an affine subspace. Thus, we can model such videos by a collection of data points from a union of subspaces. The task would be then to cluster the frames so that the frames in the same cluster correspond to the same scene or activity. We propose to apply SSC for segmentation of video frames into different clusters such that the frames in each cluster represent roughly the same scene or activity while the frames in different clusters represent scenes or activities which are up to a reasonable level distinct from each other. We propose to collect a database of videos for the purpose of temporal segmentation and evaluate the performance of the SSC algorithm as well as other clustering methods.

4.4 Dimensionality reduction and clustering of nonlinear manifolds

We consider the general case where the data are sampled from a collection of nonlinear manifolds for which we do not know the metric. We assume that for each manifold, there is a single coordinate-chart in a low-dimensional space and a mapping from this coordinate-chart to the manifold. Our goal is first to cluster the data points such that the data in the same manifold belong to the same cluster. Second, we want to recover a low-dimensional representation of the data on each manifold as well as a mapping from the low-dimensional space to the manifold.

We propose to use the SSC algorithm with possibly some modifications for the simultaneous clustering and dimensionality reduction of the data lying on multiple manifolds. As we showed in the previous section, the SSC algorithm performs remarkably well for the motion segmentation problem on a large database which consists of intersecting and non-intersecting manifolds. The reason it works for clustering of data on intersecting flat manifolds is that it chooses from all data only a few points which can linearly reconstruct the given data point. As a result, no matter of the given data point being close to or far from the intersection, the sparse coefficients come from points in the same subspace. This in fact eliminates the problems of methods such as LSA and LLMC which build a similarity matrix based on nearest neighbors of each data point. So, for points near the intersection of manifolds, they choose neighbors from different manifolds which reduces their performance.

We have conjectured and experimentally observed that if we write a point as an affine combination of all other points lying on multiple nonlinear manifolds, the coefficients with the largest values correspond to points from the same manifold and also close to the given data point. This suggests that we can construct a similarity graph from the sparse coefficients and obtain the clusters by applying spectral clustering. We propose to use the sparse coefficients in each connected component to reduce the dimensionality of the data on each manifold. This way, (1) we can cluster data lying on intersecting and non-intersecting manifolds which has not been well addressed in the literature before. In fact, methods such as LLMC and LLE can not specifically handle the case of intersecting manifolds. (2) We eliminate the problem of fixing the number of nearest neighbors which is required in methods such as LLE, and Laplacian Eigenmaps since by using the sparse representation techniques we automatically choose the neighbors. (3) The embedding of data into a lower-dimensional space follows by solving an sparse eigenvalue problem which is less costly than finding the embedding in methods such as ISOMAP. Our ongoing research aims at better understanding the behavior of the method for clustering nonlinear manifolds as well as deriving theoretical guarantees for the success of the algorithm.

We have taken three handwritten digits, $\{2, 4, 6\}$, from the MNIST database (1000 images per digit) [19] and applied the SSC algorithm to the collection of data points. We have obtained a misclassification rate of 0.6% for clustering of the digits. Having clustered the data on each manifold we have used the sparse coefficients to reduce the dimensionality of the data on each manifold. The left plot in Figure 3 shows the low-dimensional embedding for the digit 2. One can see that the horizontal axis captures the variation of the top arch while the vertical axis captures the variation of the bottom loop of the digits. More precisely, moving on the horizontal axis from left to right, the top arch becomes larger and moving on the vertical axis from bottom to top, the bottom loop becomes larger. The right hand side plot in Figure 3 shows the embedding of the three digits in a two dimensional space. Clearly, the data from different digits are well separated from each other which justifies the low misclassification rate we obtained. Part of our future research is to evaluate

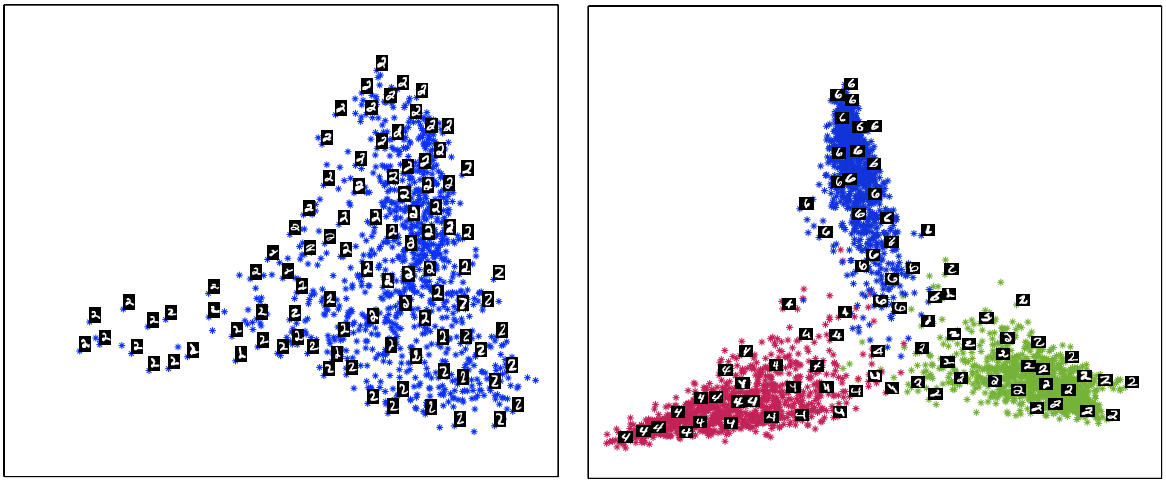

Figure 3: Left: Dimensionality reduction result for digit 2 in the MNIST database. Right: Clustering result for 3 digits in the MNIST database.

the performance of the proposed algorithm on large data sets.

References

[1] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 2008.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural Information Processing Systems*, pages 585–591, 2002.

[3] E. Candés. The restricted isometry property and its implications for compressed sensing. *C. R. Acad. Sci., Paris, Series I*, 346:589–592, 2008.

[4] E. Candés, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[5] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009.

[6] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *Int. Journal of Computer Vision*, 29(3):159179, 1998.

[7] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

[8] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, Jun 2006.

[9] D. L. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *J. Amer. Math. Soc.*, 22(1):1–53, 2009.

[10] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, October 2004.

[11] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inform. Theory*, 55(11):5302–5316, 2009.

[12] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[13] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[14] M. A. Fischler and R. C. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26:381–395, 1981.

[15] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[16] W. Hong, J. Wright, K. Huang, and Y. Ma. Multi-scale hybrid linear models for lossy image representation. *IEEE Trans. on Image Processing*, 15(12):3655–3671, 2006.

[17] I. Jollife. *Principal Component Analysis*. Springer-Verlag, 2nd edition, 2002.

[18] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE Int. Conf. on Computer Vision*, volume 2, pages 586–591, 2001.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 88(11):2278–2324, 1998.

[20] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007.

[21] S. Rao, R. Tron, Y. Ma, and R. Vidal. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[22] S. Roweis and L. Saul. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[23] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Workshop on Statistical Methods in Video Processing*, 2004.

[24] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[25] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[26] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[27] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[28] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[29] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conf. on Computer Vision*, pages 94–106, 2006.

[30] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised Segmentation of Natural Images Via Lossy Data Compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.

[31] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 287–293, 2003.