

# Sparse Subspace Clustering Algorithm of Elhamifar and Vidal

Celine Podgornik

April 2020

## 1 Introduction

The discovery of new mathematical innovations and invention of new technologies has led to a rise in the amount of data being collected in the world. In many cases, collected data is high dimensional, which can lead to problems in data analysis. These problems include algorithms having high computational times and memory requirements, leading to what is known as the “curse of dimensionality.” The “curse of dimensionality” is when the performance of an algorithm is decreased “due to the noise effect and insufficient number of samples with respect to the ambient space dimension” [2]. Converting the data to a lower dimension helps increase the performance of algorithms used in data analysis. Thus, it is often the case that data collected from many different categories lies in the union of low dimensional subspaces.

One of the many important ways that data is analyzed is using a process called subspace clustering, which separates data into the underlying low dimensional subspaces discussed above. In general, subspace clustering algorithms fall into four categories: iterative methods, algebraic approaches, statistical methods, and spectral clustering-based methods [2]. Iterative methods, for example K-subspaces and median K-flats, create a loop that assigns data points to a subspace and then fits those subspaces to a cluster. The biggest issue with taking an iterative approach is that it is necessary to know how many subspaces there are and what the dimensions of those subspaces are [2]. Algebraic approaches factor a data matrix, find the similarity matrix, and then find a segmentation using the threshold method. The drawbacks to this method include that it is strongly affected by noise and outliers, and that the subspaces must be independent. Algebraic-geometric approaches, for example Generalized Principal Component Analysis, take a polynomial, the gradient of which at some point produces a vector normal to the subspace that point is contained in, and fit the data to it. The main drawbacks to this approach are that it is strongly affected by noise and outliers, and that it has a complexity that increases exponentially as the dimensions and number of the subspaces increase [2]. Iterative statistical methods, for example Mixtures of Probabilistic PCA, create a loop that clusters the data and then uses Expectation Maximization to estimate the

subspace of a given point. This method assumes that the data has a Gaussian distribution. The biggest issues with iterative statistical methods are that the numbers and dimensions of subspaces must be known and that they are extremely affected by the initialization[2]. Spectral clustering-based methods fall into two categories: local and global. Local spectral clustering-based methods, for example Local Subspace Affinity, build similarities between pairs of points using local information about the points and then finding the segmentation of the data with spectral clustering. The drawbacks of this methods include being strongly affected by the neighborhood size and having difficulty with points near the intersection of subspaces. Global spectral clustering-based methods, such as spectral curve clustering, attempt to deal with the limitations of the local spectral clustering-based methods by using global information about the points to build similarities [2].

Ehsan Elhamifar and René Vidal developed a subspace clustering algorithm based on sparse representation. Their sparse subspace clustering algorithm takes a set of  $n$  data points from a union of linear or affine subspaces and finds the number of subspaces there are, the dimensions of the subspaces, a basis for every one of the subspaces, and which points from the data belong to which subspace (i.e. the segmentation of the data) [2]. The main idea behind this algorithm is the “self expressiveness property” of the data, or, in other words, that every point in the data set can be expressed as a combination (linear or affine) of other points in that data set [2]. In a sparse representation, a data point can be expressed as a combination of a few points from the particular subspace that it belongs to. Finding the sparse representation of subspaces is, in general, NP-hard. Therefore, Elhamifar and Vidal considered the  $\ell_1$  relaxed version of it, where, under certain conditions on the subspaces and data distribution, the  $\ell_1$ -minimization program gives the desired result using convex programming tools [1].

Subspace clustering has many applications in image processing (i.e. image representation and image compression) and computer vision (i.e. image segmentation, motion segmentation, and temporal video segmentation). In addition to these applications, sparse subspace clustering can also deal with noise, sparse outlying entries, and missing entries.

## 2 Problem Formulation

### 2.1 Sparse Optimization Problem and Clustering

The sparse subspace clustering algorithm of Elhamifar and Vidal consists of two steps: 1) for every point in the data, find a few other points that belong to the same subspace as the original point using a global sparse optimization program and 2) use the information found in the first step to figure out how the data is clustered using K-means clustering or spectral clustering [2]. First, consider some terminology. Let an arrangement of  $n$  linear subspaces of  $\mathbb{R}^D$  of dimensions  $\{d_\ell\}_{\ell=1}^n$  be denoted by  $\{S_\ell\}_{\ell=1}^n$  and let  $N$  given data points, free

of noise, lying in the union of  $n$  subspaces be denoted by  $\{y_i\}_{i=1}^N$ . These data points are contained in a matrix denoted by  $Y = [y_1, \dots, y_N] = [Y_1, \dots, Y_N]\Gamma$ , where  $Y_\ell \in \mathbb{R}^{D \times N_\ell}$  and  $\Gamma \in \mathbb{R}^{N \times N}$ . Note:  $Y_\ell$  has rank  $d_\ell$ , where  $N_\ell$  are the points in  $S_\ell$ ,  $N_\ell > d_\ell$ , and  $\Gamma$  is a permutation matrix that is not known. Then  $Y$  is used to find how many subspaces there are, the dimensions of the subspaces, a basis for every one of the subspaces, and the segmentation of the data [2].

Elhamifar and Vidal call the first step of their algorithm the “Sparse Optimization Problem.” As briefly discussed in the introduction, this step takes advantage of the “self-expressiveness property of the data” [2]. In other words, every data point  $y_i \in \cup_{\ell=1}^n S_\ell$  can be written:  $y_i = Yc_i$  (Equation 1), where  $c_i = [c_{i1} \ c_{i2} \ \dots \ c_{iN}]$  and  $c_{ii} = 0$ . Then  $Y$  is a dictionary where each point is written as a linear combination of the other points (Note:  $c_{ii} = 0$  gets rid of the solution where a point is written as a linear combination of itself). Since the dimension of a subspace is usually less than the number of data points within the subspace,  $Y$  is not generally unique. Thus, each  $Y_\ell$ , and therefore  $Y$ , has a non-trivial nullspace, meaning each data point has infinitely many depictions. Out of these infinitely many depictions, there exists a subspace-sparse representation,  $c_i$ , the nonzero entries of which correspond to data points from the same subspace as  $y_i$  and give the dimension of the underlying subspace. The solution set for the system of equations from equation 1 can be decreased in size by taking the  $\ell_q$ -norm of each solution, i.e. finding  $\min \|c_i\|_q$ . As  $q$  goes to zero, the sparsity of the of the solution increases. Elhamifar and Vidal’s sparse optimization program uses the  $\ell_1$ -norm,  $\min \|c_i\|_1$ , which can be rewritten in matrix form as  $\min \|C\|_1$  (Equation 2), where  $Y = YC$  and  $\text{diag}(C) = 0$ . Solving this sparse optimization problem gives the subspace-sparse representations of data points that is then used to figure out how the data is clustered. [2]

The second step of Elhamifar and Vidal’s algorithm, entitled “Clustering using Sparse Coefficients” [2], is to determine the how the data is clustered using the solution to equation 2. First, a weighted graph,  $G$ , is built such that the set of  $N$  nodes that correspond to the  $N$  data points is called  $V$ ,  $E$  is the set of edges between every node and is a subset of  $V \times V$ , and the weights of the edges are represented by a symmetric non-negative similarity matrix,  $W$ , [2]. Thus  $G = (V, E, W)$ . Recall that every point in the data set is represented by a subspace-sparse representation due to the sparse optimization program. Consider the data point  $y_i \in S_\ell$ . This data point can be written as a linear combination of a set of points in its subspace that includes the data point  $y_j \in S_\ell$  (i.e.  $y_j$  is in the sparse representation of  $y_i$ ). That  $y_j$ , however, might be written as a linear combination of points in its subspace that does not include  $y_i$  (i.e.  $y_i$  might not be in the sparse representation of  $y_j$ ). This leads to  $W = |C| + |C|^T$  to be the best choice for the similarity matrix of the graph  $G$ . With this choice of weight matrix, nodes  $i$  and  $j$  will be connected to each other as long as either  $y_i$  is in the sparse representation of  $y_j$ ,  $y_j$  is in the sparse representation of  $y_i$ , or  $y_i$  and  $y_j$  are in each others sparse representations. The data is then clustered into their respective subspaces by using spectral clustering or K-means clustering on the graph  $G$ . [2]

The steps of the SSC algorithm can be summarized in the following algorithm

created by Elhamifar and Vidal and presented in their 2012 article [2]:

---

**Algorithm 1:** Sparse Subspace Clustering (SSC)

---

**Input** : A set of points  $\{y_i\}_{i=1}^N$  lying in a union of  $n$  linear subspaces  $\{S_i\}_{i=1}^n$ .

1. Solve the sparse optimization problem in the case of uncorrupted data (Equation 2) or in the case of corrupted data (the equation for corrupted data can be found in [2] on page 5).
2. Normalize the columns of  $C$  as  $c_i \leftarrow \frac{c_i}{\|c_i\|_\infty}$ .
3. Form a similarity graph with  $N$  nodes representing the data points. Set the weights on the edges between the nodes by  $W = |C| + |C|^T$ .
4. Apply spectral clustering to the similarity graph.

**Output:** Segmentation of the data  $Y_1, Y_2, \dots, Y_n$ .

---

## 2.2 Graph Connectivity

As explained in Theorem 3.1, shown in the results section, the  $\ell_1$ -minimization problem finds the subspace sparse representations of data points. Therefore, data points lying in different subspaces are not connected to one another in the similarity graph formed from the sparse subspace clustering algorithm. Data points that do lie in the same subspaces, however, are connected to each other. Thus the similarity graph has  $n$  connected components [2]. In Nasihatkon and Hartley’s article entitled “Graph connectivity in sparse subspace clustering,” they showed that when the dimensions of subspaces are greater than 3 with data that has an odd distribution, it is sometimes the case that multiple graph components are formed from points in the same subspace [2]. To help combat this, Elhamifar and Vidal added the following “regularization term” (Equation 3) to their sparse optimization program to help data points within the same subspace connect [2]:

$$\|C\|_{r,0} \triangleq \sum_{i=1}^N I(\|c^i\|_2 > 0).$$

Note that the indicator function is denoted by  $I(\cdot)$  and the  $i^{th}$  row of  $C$  is denoted by  $c^i$ . This equation stems from the idea that data points will form a single component of the similarity graph if each data point chooses a few common points from its subspace to form its sparse representation [2]. Choosing the common points from the subspace of a point is done by minimizing equation 3, which is basically the same as minimizing the number of nonzero rows in  $C$ . The minimization of equation 3, however, is known to be in general NP-hard. Thus, the following convex relaxation (equation 4) is used instead [2]:

$$\|C\|_{r,1} \triangleq \sum_{i=1}^N \|c^i\|_2.$$

Consequently, solving the following equation (equation 5) increases the connectivity of data points that reside in the same subspace [2]:

$$\min \|C\|_1 + \lambda_r \|C\|_{r,1} \text{ s.t. } Y = YC, \text{ diag}(C) = 0, \text{ where } \lambda_r > 0.$$

Note that  $\lambda_r$  determines the balance between the connectivity of the graph and how sparse the solution is [2].

### 3 Results

#### 3.1 Clustering Linear Subspaces

First, consider the following notations: Let the set of  $n$  linearly independent subspaces be denoted by  $\{S_\ell\}_{\ell=1}^n$ ; Let  $\{y_i \in \mathbb{R}^D\}_{i=1}^N$  be a collection of data points from the union of  $\{S_\ell\}_{\ell=1}^n$ ; Let  $\{d_i < D\}_{i=1}^n$  be a collection of the unknown dimensions of the subspaces; Let  $\{A_i \in \mathbb{R}^{D \times d_i}\}_{i=1}^n$  be a collection of the unknown bases of the subspaces; Let  $Y_i \in \mathbb{R}^{D \times N_i}$  be the collection of  $N_i$  data points from a given subspace  $i$  [1]. Let the data matrix be denoted by  $Y = [y_1, y_2, \dots, y_N] = [Y_1, Y_2, \dots, Y_n]\Gamma \in \mathbb{R}^{D \times N}$ , due to the fact that it is as of yet unknown which data points belong to which subspaces. Note that  $N = \sum_{i=1}^n N_i$  and  $\Gamma \in \mathbb{R}^{N \times N}$  is a permutation matrix, that is not known, describing how the data is segmented [1].

As discussed in the sparse optimization problem and clustering section, the above notations lead to equation 1 (i.e.  $y_i = Yc_i$ , where  $c_i = [c_{i1} \ c_{i2} \ \dots \ c_{iN}]$  and  $c_{ii} = 0$ ) and the sparse optimization program use the  $\ell_1$ -norm,  $\min \|c_i\|_1$ , to find that the subspace-sparse representation of the data points is given by  $c = \Gamma^{-1}[c_1^T, \dots, c_n^T]$ . This leads to the following theorem (quoted from Elhamifar and Vidal's article [1] with minor notation changes):

**Theorem 3.1** *Let  $Y \in \mathbb{R}^{D \times N}$  be a matrix whose columns are drawn from a union of  $n$  independent linear subspaces. Assume that the points within each subspace are in general position. Let  $y$  be a new point in subspace  $i$ . The solution to the  $\ell_1$  problem (i.e.  $\min \|c_i\|_1$ ),  $c = \Gamma^{-1} [c_1^T, \dots, c_n^T] \in \mathbb{R}^n$  is block sparse, i.e.  $c_i \neq 0$  and  $c_j = 0$  for all  $j \neq i$ . [1]*

Proof.

Let  $c$  be a sparse representation of the data point  $y \in S_\ell$ . We know that  $c$  exists since all of the points in every subspace are in general position. So  $y = Yc$ , where  $c_i \neq 0$  and  $c_j = 0$  for all  $j \neq i$ .

Assume that  $c^*$  is a solution to the  $\ell_1$  minimization program. Then  $c^*$  is a vector of minimum  $\ell_1$  norm and  $y = Yc^*$  is satisfied by  $c^*$ .

Let the error between the optimal solution and the sparse solution be denoted by  $h = c^* - c$ . Let  $h_i$  denote the error for the points in subspace  $i$  and let  $h_j$  denote the error for the points in other subspaces. The indices of these two

vectors are disjoint subsets of the indices of  $h$  (i.e. each set of indices  $i$  and  $j$  have no elements in common). Then  $h$  can be written as the sum of these two vectors, i.e.  $h = h_i + h_j$ .

By way of contradiction, assume that  $h_j \neq 0$ . Consider the vector  $c^*$ . This vector can be written as  $c + h_i + h_j$ . Thus  $y = Yc^* = Y(c + h_i + h_j) = Y(c + h_i) + Y(h_j)$ . Now, we know that  $Y(c + h_i) \in S_\ell$  because  $y \in S_\ell$ . Furthermore, since it is assumed that the subspaces are independent,  $Y(h_j) \notin S_\ell$ . This leads to the conclusion that  $Y(h_j) = 0$ , which implies that  $y = Yc^* = Y(c + h_i)$ . Since the indices of  $h_i$  and  $h_j$  are disjoint sets,  $\|c + h_i\|_1 < \|c + h_i + h_j\|_1 = \|c^*\|_1$ . This means that  $c + h_i$  is a good solution to the  $\ell_1$  minimization program. This is a contradiction to the optimality of  $c^*$  being a solution. Thus  $h_j = 0$  and  $c_j^* = c_j = 0$ .

Therefore the solution to the  $\ell_1$  problem is block sparse. ■ [1]

The following theorem (quoted from Elhamifar and Vidal's article [1]) generalizes Theorem 3.1 for affine subspaces and its proof is similar to that of Theorem 3.1:

**Theorem 3.2** *Let  $Y \in \mathbb{R}^{D \times N}$  be a matrix whose columns are drawn from a union of  $n$  independent affine subspaces. Assume that the points within each subspace are in general position. Let  $y$  be a new point in subspace  $i$ . The solution to the  $\ell_1$  problem in (13),  $s = \Gamma^{-1} [s_1^T, \dots, s_n^T] \in \mathbb{R}^n$  is block sparse, i.e.  $s_i \neq 0$  and  $s_j = 0$  for all  $j \neq i$ . [1]*

Now consider the similarity graph  $G$ , as outlined in the sparse optimization problem and clustering section. From Theorem 3.1, it is known that the nodes of the graph that represent data points in the same subspace are all connected, whereas the nodes of the graph that represent data points that are in different subspaces are not connected. Recall that the weight on the edges of the graphs was chosen to be  $W = |C| + |C|^T$ , where  $C$  is the adjacency matrix of the graph. So  $W$  has the following form:

$$\begin{bmatrix} \sim W_1 & 0 & \dots & 0 \\ 0 & \sim W_2 & \dots & 0 \\ & & \vdots & \\ 0 & 0 & \dots & \sim W_n \end{bmatrix} \Gamma.$$

Note that  $\Gamma$  is a permutation matrix. Now, denote the Laplacian matrix of  $G$  by  $L$  and let  $D \in \mathbb{R}^{N \times N}$  denote a diagonal matrix where  $D_{ii} = \sum_j C_{ij}$ . Then  $L = D - \tilde{C}$  by definition [1].

Consider the following proposition from spectral graph theory, quoted from [1],

**Proposition 3.3** *The multiplicity of the zero eigenvalue of the Laplacian matrix  $L$  corresponding to the graph  $G$  is equal to the number of connected components of the graph. Also, the components of the graph can be determined from*

the eigenspace of the zero eigenvalue. More precisely, if the graph has  $n$  connected components, then  $u_i = [0, 0, \dots, 1_{N_i}^T, 0, \dots, 0]^T$  for  $i \in \{1, 2, \dots, n\}$  is the  $i^{th}$  eigenvector of  $L$  corresponding to the zero eigenvalue which means that the  $N_i$  nonzero elements of  $u_i$  belong to the same group.

It follows from proposition 3.3 that the number of subspaces can be estimated as the number of zero eigenvalues that  $L$  has, as determined from applying K-means to a subset of the eigenvectors of  $L$  [1].

### 3.2 Miscellaneous Subspace-Sparse Recovery Theory

The following definitions, quoted from Elhamifar and Vidal's articles ([2],[1]), are also important to the sparse subspace clustering algorithm of Elhamifar and Vidal.

**Definition 3.4** A collection of subspaces  $\{S_i\}_{i=1}^n$  is said to be independent if  $\dim(\oplus_{i=1}^n S_i) = \sum_{i=1}^n \dim(S_i)$ , where  $\oplus$  denotes the direct sum operator. [2]

**Definition 3.5** A collection of subspaces  $\{S_i\}_{i=1}^n$  is said to be disjoint if every pair of subspaces intersect only at the origin. In other words, for every pair of subspaces we have  $\dim(S_i \oplus S_j) = \dim(S_i) + \dim(S_j)$ . [2]

**Definition 3.6** The smallest principal angle between two subspaces  $S_i$  and  $S_j$ , denoted by  $\theta_{ij}$  is defined as,

$$\cos(\theta_{ij}) \triangleq \max_{v_i \in S_i, v_j \in S_j} \frac{V_i^T v_j}{\|v_i\|_2 \|v_j\|_2}.$$

[2]

**Theorem 3.7** Consider a collection of data points drawn from  $n$  independent subspaces  $\{S_i\}_{i=1}^n$  of dimensions  $\{d_i\}_{i=1}^n$ . Let  $Y_i$  denote  $N_i$  data points in  $S_i$ , where  $\text{rank}(Y_i) = d_i$ , and let  $Y_{-i}$  denote data points in all subspaces except  $S_i$ . Then, for every  $S_i$  and every nonzero  $y$  in  $S_i$ , the  $\ell_q$ -minimization program

$$\begin{bmatrix} c^* \\ c_-^* \end{bmatrix} = \left\| \begin{bmatrix} c \\ c_- \end{bmatrix} \right\|_q \text{ s.t. } y = [Y_i \quad Y_{-i}] \begin{bmatrix} c \\ c_- \end{bmatrix},$$

for  $q < \infty$ , recovers a subspace-sparse representation, i.e.,  $c^* \neq 0$  and  $c_-^* = 0$ . [2]

Note: Let (equation 6) denote the following:

$$\forall x \in S_i \cap (\oplus_{j \neq i} S_j), x \neq 0 \Rightarrow \|a_i\|_1 < \|a_{-i}\|_1,$$

where  $a_i$  is the optimal solution to the  $\ell_1$ -minimization from a subspace  $S_i$  and  $a_{-i}$  is the optimal solution to the  $\ell_1$ -minimization program for all subspaces except for  $S_i$ .

**Theorem 3.8** Consider a collection of data points drawn from  $n$  disjoint subspaces  $\{S_i\}_{i=1}^n$  of dimensions  $\{d_i\}_{i=1}^n$ . Let  $Y_i$  denote  $N_i$  data points in  $S_i$ , where  $\text{rank}(Y_i) = d_i$ , and let  $Y_{-i}$  denote data points in all subspaces except  $S_i$ . The  $\ell_1$ -minimization

$$\begin{bmatrix} c^* \\ c_-^* \end{bmatrix} = \left\| \begin{bmatrix} c \\ c_- \end{bmatrix} \right\|_1 \text{ s.t. } y = [Y_i \quad Y_{-i}] \begin{bmatrix} c \\ c_- \end{bmatrix},$$

recovers a subspace-sparse representation of every nonzero  $y$  in  $S_i$ , i.e.,  $c^* \neq 0$  and  $c_-^* = 0$ , if and only if (equation 6) holds. [2]

Note: Let (equation 7) denote the following:

$$\begin{bmatrix} c^* \\ c_-^* \end{bmatrix} = \left\| \begin{bmatrix} c \\ c_- \end{bmatrix} \right\|_1 \text{ s.t. } y = [Y_i \quad Y_{-i}] \begin{bmatrix} c \\ c_- \end{bmatrix}.$$

**Theorem 3.9** Consider a collection of data points drawn from  $n$  disjoint subspaces  $\{S_i\}_{i=1}^n$  of dimensions  $\{d_i\}_{i=1}^n$ . Let  $W_i$  be the set of all full-rank submatrices  $\tilde{Y}_i \in \mathbb{R}^{D \times d_i}$  of  $Y_i$ , where  $\text{rank}(Y_i) = d_i$ . If the condition

$$\max_{\tilde{Y} \in W_i} \sigma_{d_i}(\tilde{Y}_i) > \sqrt{d_i} \|Y_{-i}\|_{1,2} \max_{j \neq i} \cos(\theta_{ij})$$

holds, then for every nonzero  $y$  in  $S_i$ , the  $\ell_1$ -minimization in (equation 7) recovers a subspace-sparse solution, i.e.,  $c^* \neq 0$  and  $c_-^* = 0$ . [2]

## 4 Conclusion

Subspace clustering is a very important method with respect to analyzing the vast amounts of data being collected in the world today. The high dimensionality of data can give rise to performance errors, however, which leads to the creation of methods that reduce data sets to lower dimensions. This improves the performance of algorithms used to analyze data, but it can render the data hard to read. Subspace clustering methods create a way to figure out which subspaces the data originally belonged to. In particular, Elhamifar and Vidal developed a method called the sparse subspace clustering (SSC) algorithm. The SSC algorithm takes  $n$  data points from a union of linear or affine subspaces and finds the number of subspaces there are, the dimensions of the subspaces, a basis for every one of the subspaces, and the segmentation of the data (i.e. which data points belong to which subspaces). This algorithm is in general NP-hard, therefore Elhamifar and Vidal consider a  $\ell_1$  relaxed version of it which gives them the desired results. An advantage to the SSC algorithm over other subspace clustering algorithms is that it can take noise, sparse outlying entries, and missing entries into account. The SSC algorithm uses many tools known in linear algebra such as similarity graphs and graph connectivity, K-means clustering, spectral clustering, and minimization.

One way that this project could be improved by future research is adding in some sections that describe and experiment with the code involved with sparse



subspace clustering. An overview of the sparse subspace clustering algorithm is given in the problem formulation section of this project, but is not expanded upon. The output of this code used on real data could be added to the results section. This addition could be tied into the applications of the sparse subspace clustering algorithm that were briefly mentioned in the introduction of this project, image processing and computer vision. In [2], for example, Elhamifar and Vidal experiment with the sparse subspace clustering algorithm with respect to segmenting multiple motions in videos and clustering images of human faces and compare its performance to the performance of other subspace clustering algorithms such as latent semantic analysis (LSA), shared computer clustering (SCC), low-rank representation (LRR), and low-rank subspace clustering (LRSC). Their experiments showed that their algorithm was superior to the other algorithms. Another way that this project could be improved by further research is by adding in proofs of the other theorems mentioned in the clustering linear subspaces and miscellaneous subspace-sparse recovery sections, as well as expanding upon the sparse subspace clustering algorithm with respect to affine subspaces. The idea that the SSC algorithm can deal with noise, sparse outlying entries, and missing entries was only briefly touched on in the introduction, so more research could be done on this. In fact, Elhamifar and Vidal stated in [2] that they were doing more research on the theoretical analysis of the presence of noise, sparse outlying entries, and missing entries in data sets. One of the limitations of Elhamifar and Vidal's sparse subspace clustering algorithm is that the performance is not optimal when it comes to very large data sets. In [2], Elhamifar and Vidal state they want to figure out how to make the process of solving the sparse optimization program and clustering work better for very large sets of data.

## References

- [1] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.
- [2] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *arXiv preprint arXiv:1203.1005*, 2012.