Subspace Segmentation Problem

Augusto Santos Jo˜ao Xavier
Institute for Systems and Robotics
IST
Lisbon

Abstract

Segmenting a mixture of linear subspaces in the presence of noise and outliers is a challenging problem which finds direct applications in computer vision, e.g., motion segmentation, structure-from-motion and object recognition.

We present two optimization formulations for the subspace segmentation problem together with the associated numerical algorithms to solve them.

In the first formulation, the segmentation problem is cast as the maximization of a convex function over a compact convex set. We propose a monotone optimization algorithm which loops between a Minimization-Maximization (MM) routine and a Escape from only-Local Maximizers (ELM) routine. The MM routine locates a local maximizer and ELM tries to escape from it towards the global one, re-initializing the MM routine when successful. For the MM routine, convergence is proved to be achieved in a finite number of steps (for a certain cost function). Our technique shows a competitive performance with both synthetic and real data image sequences, which includes the hard articulated scenario. The main drawback is that the number of subspaces to be segmented and their dimensions are required beforehand.

In the second formulation, the segmentation problem is approached through successive minimization of a \( l_1 \)-norm based convex function over the unit-norm sphere. The goal is to construct, recursively, the orthogonal complements of the subspaces. Prior knowledge of the number of subspaces or their dimensions is not required. To handle the optimization problems over the unit-sphere, we present a locally-convergent optimization algorithm with quadratic rate of convergence (Newton method on a Riemannian manifold). The performance of our approach is tested through computer simulations.

Finally, we introduce a novel theoretical framework to analyse the subspace segmentation problem. This is based on optimality criteria for convex optimization and subdifferential calculus. This framework shows that, in the absence of outliers, the subspace segmentation problem can be solved by a certain random linear program, with probability one. Incorporation of outliers lowers the probability of success, but our framework enables its theoretical computation and offers an intuitive geometrical interpretation.

1. Introduction

Today, a huge amount of videos are made to record events and study the structure of the world. Many applications require methods for segmenting these image sequences into meaningful events or scenes for easy access, analysis or editing [1].

Humans can easily discriminate independently moving objects without retrieving their specific identities. It roughly suggests that motion is an important clue for segmenting objects.

The first step for discriminating different moving objects in a sequence of images is the tracking process. Essentially, it consists in detecting specific features of the objects in each frame and arrange them into trajectories across the images.

In fact, several local features arises from the object’s movement. For instance, assume a sequence of images with the same background and one object moving around. Assume that from one frame to the next the edges of the object shift by a few pixels. Subtracting these two consecutive frames, the result is a frame in which the pixels differing significantly from zero are those ones corresponding to the aforementioned edges. Actually, we are interested in more local features. Neither edges nor lines, but specific points in the objects so that we can reconstruct their rigid structure.

There are several techniques and criteria to detect features in a sequence of images. However, these are out of the scope of this work. For more details, see [1].

The tracking process even when successfully done is just the beginning of the story. Once the features’ trajectories are produced, it is necessary to figure out which trajectories belong to which objects, so that we can process the information concerning each object separately.

Segmenting a moving object in a sequence of images, given the tracked features’ trajectories, is equivalent to identifying an associated 4-dimensional or 3-dimensional sub-
space living in a certain Euclidean ambient space. That is, each subspace identified conveys trajectory points regarding each object. This fact was firstly pointed out by Costeira and Kanade [2]. This is the principle behind most of currents works on motion segmentation. Kanatani [3] provided a simple proof of this paradigm which is not a general proof since it assumes the simplest camera model (orthographic), but this fact holds for all affine camera models [4].

2 Contributions

• A main contribution of this thesis is the formulation of the subspace segmentation as the maximization of a convex function over a compact convex set. This special structure allowed the derivation of efficient locally-convergent numerical algorithms.

• We proved the convergence of one of the proposed algorithms in a finite number of steps. The line of proof can be exploited in the future to prove the convergence of the other proposed algorithms.

• We also introduce a novel approach to tackle the subspace segmentation problem. This is based on optimality criteria for convex optimization and subdifferential calculus. This approach shows that, in the absence of outliers, the subspace segmentation problem can be solved by a certain random linear program. Incorporation of outliers lowers the probability of success, but our framework allows us to compute it beforehand and offers an intuitive geometrical interpretation.

3. Subspace Segmentation - Orthogonal Subspaces

3.1 Data Model

First of all, we shall define the concept of outlier, to be used throughout this work.

**Definition.** Let $X$ denote a finite set of points in $\mathbb{R}^n$. Given an integer $h$, an element $p$ of the set $X$ is called an outlier in $X$ if there is no a sum decomposition $\mathbb{R}^n = L_1 \oplus \ldots \oplus L_d$ with pre-defined dimensions $\dim L_i = n_i$ s.t.:

$$\#(X \cap L_i) \geq h \text{ and } \exists_i : p \in L_i$$

(1)

where $\#(S)$ denotes the cardinality of the set $S$.

In other words, assuming that part of the points of $X$ are distributed into $d$ subspaces, $p$ is an outlier in $X$ if it does not lie in no one of these nontrivial subspaces containing $h$ or more points of $X$. Of course, the definition is attached to the value of $h$. The set of outliers in $X$ will be denoted by $O_X$.

Furthermore, we shall consider a subspace to be activated if it contains at least $h$ elements of $X$. Thus, the threshold $h$ has to be settled up *a priori* in order to make sense of this terminology.

Basically, our aim in this work is to identify activated subspaces. Fig. 1 illustrates the rough idea.

In this subsection we formulates the subspace segmentation problem in the absence of outliers assuming mutually orthogonal activated subspaces. It is worth to mention that, when there is no outliers, there is no loss of generality in treating the subspace segmentation problem assuming orthogonal subspaces. That is, if we are dealing with the oblique case, we can just pre-process the data through an SVD in order to recover the orthogonal structure (proof is provided in [5]).

**Problem Formulation.** Let $X = \{x_i \in \mathbb{R}^n : i = 1, \ldots, m\}$ be a given data set s.t. $O_X = \emptyset$. Consider that there exists a direct sum decomposition $\mathbb{R}^n = L_1 \oplus \ldots \oplus L_d$ respecting:

$$\#(L_i \cap X) \geq h$$

(2)

and dim $L_i$ are known, for all $i$. Find the subspaces $L_i$, for all $i$.

In this section, we will assume without loss of generality that $d = 2$ and that the points $x_i$ lie over the unit-sphere. Indeed, if a set of points spans a subspace $L$, so do their unit-norm scaled versions. The fact that there is no loss of generality in considering $d = 2$ is in the sense that if we find two orthogonal subspaces $L_1$ and $L_2$ s.t. $\mathbb{R}^n = L_1 \oplus L_2$ then the remaining subspaces live within this couple of subspaces. For more details consult [5].

Just as a note, we will only use the $n_1$-truncated SVD throughout this thesis, where $n_1$ is the rank of the entry matrix of the SVD. That is, given $X$ s.t. $\text{rank}(X) = n_1$, then
we will just consider the first $n_1$ columns of $U$ and $V$ in the representation of the SVD of $X$.

Finding the two activated subspaces $L_1$ and $L_2$ is equivalent to solve the problem:

$$\arg\min_{Q=[Q_1 \ Q_2] \in \mathcal{O}(n)} f(Q) := \sum_{i=1}^{m} \min \{ \|Q_1^T x_i\|^2, \|Q_2^T x_i\|^2 \}$$

where $L_1 = \text{span}\{Q_1^*\}$ and $L_2 = \text{span}\{Q_2^*\}$, assuming $Q^* = [Q_1^* \ Q_2^*]$ as solution. Also $\mathcal{O}(n)$ represents the orthogonal matrix group of order $n$:

$$\mathcal{O}(n) = \{ Q \in \mathbb{R}^{n \times n} : Q^T Q = I_n \}$$

Note that neither $f(Q)$ nor $\mathcal{O}(n)$ are convex objects, but we can reformulate such a problem as a maximization problem s.t. both the objective function and the constraint set are convex objects. This is explained in theorem 3.1 below. Before proceeding to the theorem, we recall the concepts of convex combination and convex hull.

**Definition.** A convex combination of the elements $x_1, \ldots, x_m$ in $\mathbb{R}^n$ is an element of the form:

$$x = \sum_{i=1}^{m} \alpha_i x_i$$

where $\sum_{i=1}^{m} \alpha_i = 1$ and $\alpha_i \geq 0$ for $i = 1, \ldots, m$.

**Definition.** Given a set $S$, the convex hull of $S$ (denoted by $\text{co } S$) is the set of all convex combinations of elements of $S$.

Another equivalent and more intuitively definition of convex hull [6] is:

$$\text{co } S = \cap \{ C : C \text{ is convex and contains } S \}$$

Figure 2: Illustration of the convex hull of a set. The resulting set $\text{co } S$ is the “smallest” convex set containing $S$.

We now present the theorem (its proof is given in [5]).

**Theorem 3.1.** Let $X = [x_1 \cdots x_m]$ be the data matrix, $Q = [Q_1 Q_2]$, and $C = \text{co } \text{Proj}_n(n_1)$, then:

$$\min_{Q \in \mathcal{O}(n)} \sum_{i=1}^{m} \min \{ \|Q_1^T x_i\|^2, \|Q_2^T x_i\|^2 \} \Leftrightarrow \max_{\Pi \in C} \sum_{i=1}^{m} \left| (x_i^T \Pi x_i) - \frac{1}{2} \right|$$

in which $\text{co } \text{Proj}_n(n_1)$ is the convex hull of the orthogonal projectors set of order $n$ and rank $n_1$.

We are assuming in theorem 3.1 that those two programs are equivalent in the sense that if $Q^* = [Q_1^* Q_2^*]$ is a solution of the first problem, then $\Pi^* = Q_1^* Q_1^T$ is solution of the second one and if $\Pi^* = Q_1^* Q_1^T$ is solution of the second problem then $Q^* = [Q_1^* Q_2^*]$ is solution of the first one.

Such theorem allows us to obtain a more compact and structurally attractive program with cost function:

$$f_{tr}(\Pi) = \sum_{i=1}^{m} \phi_{tr}(x_i^T \Pi x_i) \in \text{Conv } \mathbb{R}^{n \times n}$$

where $\phi_{tr} : \mathbb{R} \rightarrow \mathbb{R}$ is defined as $\phi_{tr}(s) = |s - \frac{1}{2}|$ (see its graph in Fig. 3) and with a restriction set that is convex and compact (convexity and compactness are shown in [5]). Therefore, without loss of optimality, it is sufficient to restrict our attention to the subset of extreme points of $C$.

Figure 3: Graph of the function $\phi_{tr}(s) = |s - \frac{1}{2}|$.

**Definition.** The point $x \in C$ is an extreme point of $C$ when there are no two distinct points $x_1$ and $x_2$ in $C$ and no $\alpha \in (0,1)$ such that:

$$x = \alpha x_1 + (1-\alpha)x_2.$$  (8)

We denote the set of extreme points of $C$ by $\text{ext } C$.

Fig. 4 illustrates this concept.

We can prove that $\text{Proj}_n(n_1) = \text{ext } (\text{co } \text{Proj}_n(n_1))$ (see [5]) and therefore, we can restrict attention to the set $\text{Proj}_n(n_1)$, that is, we will be seeking for projectors with a pre-defined rank.

Note that $f_{tr}$ (7) is nonsmooth. The gradient of $f_{tr}$ does not exist at every point. However, a generalization of it, the subdifferential does. The subdifferential is a crucial concept which comes up in algorithms for nonsmooth convex optimization.

**Definition.** The subdifferential of $f \in \text{Conv } \mathbb{R}^{n}$ at the point $x$ is the set of vectors $s \in \mathbb{R}^{n}$ which satisfy:

$$f(y) \geq f(x) + (s, y-x)$$

for all $y \in \mathbb{R}^{n}$.  (9)
It is denoted by $\partial f(x)$. An element $s \in \partial f(x)$ is called a subgradient of $f$ at $x$.

Fig. 5 highlight an important fact derived from the definition. Assume we are maximizing a convex function $f$ subject to the restriction set $C$. By choosing a subgradient $s$ in $\partial f(x)$, we see that any point $\hat{y}$ of $C$ in the half-space $\{y \in \mathbb{R}^n : \langle s, y - x \rangle > 0\}$ beats the point $x$. Specifically, returning to our case, given $\Pi_k \in C$, it follows that:

$$\langle S, Y - \Pi_k \rangle > 0 \Rightarrow f(Y) > f(\Pi_k)$$  \hspace{1cm} (10)

for any $S \in \partial f(\Pi_k)$ and $Y \in C$.

In other words, by local analysis, we can predict one direction $D$ in which the function always grows. The subdifferential here allows us to broaden this local property even to nonsmooth (but convex) functions. In the next section we will explore this statement in more detail.

### 3.2 MM routine

Based on the formulation developed in the previous subsection, we are dealing with the following optimization program:

$$\max_{\Pi \in C} f(\Pi),$$  \hspace{1cm} (11)

in which $C$ is convex compact and $f \in \text{Conv} \mathbb{R}^{n \times n}$. The problem as stated is hard. Nevertheless, we can explore the convex structure of the objective function and derive a single Minimization-Maximization (MM) routine. Indeed, $f$ can be lowerbounded by an affine function at any chosen point. It touches this lower bound at the given point. Specifically:

$$\forall S \in \partial f(\Pi_0) : f(\Pi) \geq f(\Pi_0) + \langle S, \Pi - \Pi_0 \rangle$$  \hspace{1cm} (12)

That is, we can construct an iterative routine, which we call MM, that iteratively increases the objective function by solving a sequence of single linear programs:

$$\Pi_{k+1} \in \arg \max_{\Pi \in C} f(\Pi_k) + \langle S_k, \Pi - \Pi_k \rangle$$  \hspace{1cm} (13)

where $S_k \in \partial f(\Pi_k)$

The MM routine for the cost function $f_{tr}$ is presented in table 1 (see [5] for more details in the implementation). A more top level presentation of the routine coupled with the ELM is presented in Fig. 6.

### 3.3 MM-ELM routine

The previous routine is responsible to return local maximizers. Since there is noninteresting local maximizers, we shall create an heuristic to escape from these maximizers towards the global ones.

Some observations of the qualitative behavior of the MM routine with $f_{tr}$ suggest that given a maximizer $\Pi^*$, probably, those points $x_i$ with lower or greater values of $h(x_i) = x_i^T \Pi^* x_i$ (call it dominant points) could be in fact in one of the subspaces $L_1^+$ or $L_1$, respectively. Then, as an heuristic, we can construct a new projector $\Pi_{\text{new}}$ onto the subspace spanned by the points with greater or lower value...
Table 1: MM routine: $n_1$ represents the dimension of one of the (two pairwise orthogonal) subspaces, $X$ is the data matrix, $\Pi_k$ is the $k$th projector in the sequence defined by the algorithm and $\Pi^*$ is the local maximizer.

<table>
<thead>
<tr>
<th>In:</th>
<th>$n_1$, $X$</th>
</tr>
</thead>
<tbody>
<tr>
<td>step 1)</td>
<td>Evaluate the $n_1$-truncated SVD of $X$: $X = U \Sigma V^T$</td>
</tr>
<tr>
<td>step 2)</td>
<td>$k \leftarrow 0$, $\Pi_0 \leftarrow UU^T$</td>
</tr>
<tr>
<td>step 3)</td>
<td>Construct $G_k = \sum_{i=1}^{m} \alpha_i x_i x_i^\top$ where:</td>
</tr>
<tr>
<td></td>
<td>$\alpha_i = \begin{cases} 1, &amp; \text{if }</td>
</tr>
<tr>
<td>step 4)</td>
<td>Check if $\Pi_k$ satisfies the necessary condition of local maximizer ($\partial f(\Pi_k, X) \subset N_C(\Pi_k)$):</td>
</tr>
<tr>
<td></td>
<td>$\sum_{i=n-r+1}^{n} \lambda_i(G_k) - (G_k, \Pi_k) &lt; 0$</td>
</tr>
<tr>
<td>If yes,</td>
<td>$\Pi^* \leftarrow \Pi_k$ and go to out. If not, go to 5).</td>
</tr>
<tr>
<td>step 5)</td>
<td>If $k &gt; 0$, construct $G_k = \sum_{i=1}^{m} \alpha_i x_i x_i^\top$, where:</td>
</tr>
<tr>
<td></td>
<td>$\alpha_i = \begin{cases} 1, &amp; \text{if }</td>
</tr>
<tr>
<td>else,</td>
<td>continue to step 6).</td>
</tr>
<tr>
<td>step 6)</td>
<td>Choose $\Pi_{k+1}$ such that</td>
</tr>
<tr>
<td></td>
<td>$\Pi_{k+1} \in \arg \max_{\Pi \in C} (G_d, \Pi)$</td>
</tr>
<tr>
<td>Do $k \leftarrow k + 1$ and jump to 3).</td>
<td></td>
</tr>
<tr>
<td>Out:</td>
<td>$\Pi^* \in \text{Proj}_n(n_1)$ (local maximizer)</td>
</tr>
</tbody>
</table>

of $h(x_i)$, re-initializing the MM with this new projector if $f(\Pi_{\text{new}}) > f(\Pi^*)$.

To sum up, we can sort the points, based on its values of $h(x_i)$, in increasing order and decreasing order. Then, we can construct a projector onto $\mathcal{L}_{\text{new}} = \text{span}\{x_1, \ldots, x_{n_1} : h(x_1) \leq \ldots \leq h(x_{n_1})\}$. A sketch of the routine is illustrated in Fig. 7.

Figure 7: Flowchart sketching the MM-ELM loop.

### 3.4 Log-version of $f_{tr}$

Our aim is to build up an algorithm to achieve a global solution in presence of outliers and for a general configuration of the activated subspaces. Thus, a necessary condition for a good performance is that the stronger maximizers of the objective function should codify the activated subspaces. In fact, this does not hold for $f_{tr}$ when the subspaces are oblique. In this configuration, the maximizers of $f_{tr}$ are displaced from the interesting solutions and this displacement depends on the angle of the subspaces.

To bypass this default, we can propose another cost function $\phi(\cdot)$ to replace $\phi_{tr}(s) = |s - \frac{1}{2}|$. We will consider the function $\phi_{\log} : [0, 1] \rightarrow \mathbb{R}$ defined by $\phi_{\log}(s) = \log(s(1 - s) + \varepsilon)$ and therefore, our new cost function is given by:

$$f_{\log}(\Pi) = \sum_{i=1}^{m} \phi_{\log}(x_i^\top \Pi x_i) = \sum_{i=1}^{m} \phi_{\log}((x_i^\top \Pi x_i)(1 - x_i^\top \Pi x_i) + \varepsilon)$$

In fact, the maximizers of this new cost function for any arbitrary configuration of the activated subspaces and with
a huge amount of outliers lies over the interesting solutions (see [5]).

### 3.5 Results

Next we present some relevant results. The terminology "\( \mathcal{L}_i \)" in the head of the tables means the number of points of the data matrix \( X \) that lies in \( \mathcal{L}_i \). Also, \( O_X \) represents the number of outliers and "Rel." means the reliability of the algorithm in achieving the activated subspaces.

- **Two Oblique Subspaces.**

<table>
<thead>
<tr>
<th>Two Oblique Subspaces</th>
<th>( O_X )</th>
<th>( \mathcal{L}_1 )</th>
<th>( \mathcal{L}_2 )</th>
<th>Rel. ( f_{\log} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>35</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>25</td>
<td>25</td>
<td>99.63</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>20</td>
<td>20</td>
<td>97.98</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>15</td>
<td>15</td>
<td>83.45</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>13</td>
<td>12</td>
<td>69.87</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>10</td>
<td>10</td>
<td>32.63</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: MM-ELM routine with \( f_{\log} \). Two 4-dimensional oblique activated subspaces.

- **Two Oblique Subspaces with nontrivial intersection.**

<table>
<thead>
<tr>
<th>Nontrivial Intersection</th>
<th>( O_X )</th>
<th>( \mathcal{L}_1 )</th>
<th>( \mathcal{L}_2 )</th>
<th>Rel. Nontrivial</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>35</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>25</td>
<td>25</td>
<td>97.24</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>20</td>
<td>20</td>
<td>77.06</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>15</td>
<td>15</td>
<td>34.51</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Two 4-dimensional oblique subspaces subject to \( \dim \mathcal{L}_1 \cap \mathcal{L}_2 = 1 \). Cost function used was \( f_{\log} \).

For results with noise and with real data, consult [5].

### 4 Subspace Segmentation - General Formulation

In this section we adopt the model:

\[
\min_{\|u\| = 1} \|X^T u\|_0, \quad (16)
\]

in order to characterize the subspace segmentation problem. Actually we shall not handle 16 since such a problem is combinatorial. A good heuristic to this problem is:

\[
\min_{\|u\| = 1} \|X^T u\|_1, \quad (17)
\]

in a sense that is explained in [5].

Our aim is to tackle (17) through two algorithms.

The first algorithm named Newton-sphere (NS), minimizes a generic smooth function restricted to the unit-sphere. The NS algorithm is initialized from many different points over one of the hemispheres of the sphere (the objective function is symmetric). The best minimizer is then selected.

The second algorithm NS-ELM arises from the fact that the NS is usually trapped in local minimizers which are not usually interesting. The NS-ELM algorithm is constructed through an adaptation of the ELM routine\(^1\). For both the NS and the NS-ELM, it is necessary to evaluate the second order derivative of the objective function which in the case (17) is not even differentiable. This fact lead us to work with a smooth (approximate) version of the cost function \( f(u) = \|X^T u\|_1 \), namely:

\[
\min_{\|u\| = 1} f_{sq} := \sum_{i=1}^{m} \sqrt{(x_i^T u)^2 + \epsilon} \quad (18)
\]

We will also consider another cost function for (17). The idea is the same as in previous chapter: construct a log-version of \( \|X^T u\|_1 \), which is expected to turn the algorithm more robust to outliers:

\[
\min_{\|u\| = 1} f_{\logns} := \sum_{i=1}^{m} \log((x_i^T u)^2 + \epsilon) \quad (19)
\]

### 4.1 Recursive Strategy

By solving (17), and if it succeeds, we obtain a solution \( u^* \) that is orthogonal to a \( (n - 1) \)-dimensional hyperplane containing activated subspaces. In order to achieve such subspaces (with arbitrary dimensions) it is necessary to reduce the dimension of the ambient space following a recursive strategy as explained in what follows.

As an example, assume we have to identify two straight lines in \( \mathbb{R}^3 \) as depicted in Fig. 8.

Denote the data matrix as \( X_1 \). The algorithm would start by solving:

\[
\min_{\|u\| = 1} \|X_1^T u\|_0, \quad (20)
\]

and if it succeeds it will find the solution \( u^* \) s.t. \( \text{span}\{u^*\} = \mathcal{L}_1^\bot \). This corresponds to a first level of recursion.

\(^1\)In this section this acronym shall be read as escape from only-local minima.
Up to here, we have a solution pointing out a subspace \( \mathcal{L}_1 \) that eventually contains more activated subspaces (in this case it contains two straight lines). Then, we could construct a new data matrix \( X_2 \) as follows: i. Evaluate the \((n - 1)\)-truncated SVD of the matrix \( X_T \) consisting of those points of \( X_1 \) lying in \( \mathcal{L}_1 \), yielding \( X_T = U \Sigma V^\top \), ii. Define \( X_2 = V^\top \).

Note that if we try to solve the problem without reducing the dimensionality of the ambient space:

\[
\min_{\|u\| = 1} \|X_1^\top u\|_0, \tag{21}
\]

then, the global solution would be the same as before: \( u^* \) s.t. \( \text{span}\{u^*\} = \mathcal{L}_1^1 \).

Thus, we can transport the problem of identifying \( \mathcal{L}_{11} \) and \( \mathcal{L}_{12} \) in \( \mathbb{R}^3 \) to a subspace of dimension \( \text{dim} \mathcal{L}_1 = 2 \).

We should solve:

\[
\min_{\|u\| = 1} \|X_2^\top u\|_0, \tag{22}
\]

where \( u \) is now a 2-dimensional vector and \( X_2 := V^\top \). This corresponds to the second level of recursion.

Hence, through the possible solutions in (22) we can obtain the set of points lying in one of the straight lines \( \mathcal{L}_{11} \) or \( \mathcal{L}_{12} \). These points are labeled by a set of indexes \( \mathcal{I} \) s.t. \( \langle x_i, u^* \rangle = 0 \ \forall i \in \mathcal{I} \).

Once we achieve a straight line (suppose the red one in Fig. 8), we cannot go deeper in the recursion, then we try to peel all possible subspaces within this level of recursion (second level so far). That is, we can construct a matrix \( X_2^\top \) corresponding to those points of \( X_2 \) that do not lie over the red line. We can solve:

\[
\min_{\|u\| = 1} \|X_2^e^\top u\|_0, \tag{23}
\]

in order to find the green straight line. By finding this subspaces in \( \mathbb{R}^2 \) we can identify the activated subspaces in \( \mathbb{R}^3 \) through the map \( U \Sigma V^\top \).

To conclude, we can proceed recursively with this strategy up to find all activated subspaces. This recursion stops when there is no more subspaces with more than \( h \) points.

### 4.2 NS-ELM Routine

Since the restriction set is not convex, local minimizers are to be expected. As in section 3, there are minimizers that represent no interesting solutions. Thus, the ELM routine from section 3 has been adapted to this case.

The idea for the ELM routine here is: given \( u^* \) a local minimizer, we can sort the points of \( X \) in increasing order of their values \( |\langle x_i, u^* \rangle| \). Then, we can construct a vector \( u_{new} \in \text{span}\{x_1, \ldots, x_k\} \). This new point works as a re-initialization for the NS if \( f_{obj}(u_{new}) < f_{obj}(u^*) \). If this latter inequality does not hold, then we can construct \( x_{new} \in \text{span}\{x_1, \ldots, x_{k+1}\} \) and so on, for the same reason presented in section 3.

### 4.3 Results

We present some relevant results of the NS-ELM with cost function \( f_{\logns} \). We remark that the algorithm is initialized from 60 points. The best minimizer is then selected.

- **Two Oblique Subspaces.**

<table>
<thead>
<tr>
<th>( O_X ) [%]</th>
<th>( L_1 ) [%]</th>
<th>( L_2 ) [%]</th>
<th>Rel.NS-ELM.log[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>35</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>25</td>
<td>25</td>
<td>99.1</td>
</tr>
<tr>
<td>60</td>
<td>20</td>
<td>20</td>
<td>88</td>
</tr>
<tr>
<td>70</td>
<td>15</td>
<td>15</td>
<td>70.5</td>
</tr>
</tbody>
</table>

Table 4: Comparison: NS-ELM vs NS. Two 4-dimensional oblique subspaces.

- **Two oblique subspaces with nontrivial intersections.**

For additional results see [5]. The NS-ELM is much more sensitive to noise than the MM-ELM and moreover the MM-ELM is much more efficient since it is initialized at only one point. In fact, the NS-ELM can take up to 2 minutes to solve the problem for two subspaces in some configurations whereas the MM-ELM spends about 3 seconds in average.
Table 5: NS-ELM.log. Two 4-dimensional oblique subspaces subject to $\dim L_1 \cap L_2 = 1$

<table>
<thead>
<tr>
<th>$O_1$ [%]</th>
<th>$L_1$ [%]</th>
<th>$L_2$ [%]</th>
<th>Rel.NS-ELM.log [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>35</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>25</td>
<td>25</td>
<td>93.4</td>
</tr>
<tr>
<td>60</td>
<td>20</td>
<td>20</td>
<td>68.75</td>
</tr>
<tr>
<td>70</td>
<td>15</td>
<td>15</td>
<td>23.3</td>
</tr>
</tbody>
</table>

5 Subspace Segmentation - Linear Program

In the absence of outliers in the data matrix $Y = [y_1 \cdots y_m]^\top$, the solutions of:

$$\min_c \|Y u\|_1 \quad (24)$$

are included in the solution set of:

$$\min \|u\| \quad (25)$$

for almost all choices of $c$. Thus, the combinatorial problem (25) can be solved by a single linear program 24. Moreover, any solution of (24) codifies an activated subspace. Varying $c$, all of the interesting solutions can be achieved.

The optimality condition supporting this claim comes from the convex analysis. We know that for a convex program:

$$\min_{x \in C} f(x) \quad (26)$$

with $f \in \text{Conv } \mathbb{R}^n$ and $C$ a closed convex set, we have that (see [6], chap. 7):

$$0 \in \partial f(x^*) + N_C(x^*) \quad (27)$$

in which $x^*$ represents a solution of (26). Condition (27) can be rewritten as:

$$(-\partial f(x^*)) \cap N_C(x^*) \neq \emptyset \quad (28)$$

Equation (28) represents the condition behind the theory presented in this section.

We will present a pictorial proof (through an example) of this fact rather than a strict mathematical proof. We will be looking to the graph of $\partial \|Y u\|_1$ in order to extract minimality conditions.

We will also interpret geometrically (through convex analysis) the effect of adding outliers in the system.

5.1 Example Without Outliers

Let $f(x) = \|Y x\|_1$, where:

$$Y = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

(29)

In this case we have to identify two subspaces: $L_1 = \text{span}\{e_1\}$ and $L_2 = \text{span}\{e_2\}$, where $e_1 = (1, 0)$ and $e_2 = (0, 1)$ represent the canonical vectors of $\mathbb{R}^2$. There holds $f(x) = 4|e_1^\top x| + 2|e_2^\top x|$. So:

$$\partial f(x) = 4\partial |e_1^\top x| + 2\partial |e_2^\top x| \quad (30)$$

Moreover, we have $\partial y_i^\top x = \text{sgn}(y_i^\top x)y_i$, where:

$$\text{sgn}(p) = \begin{cases} 1 & \text{if } p > 0 \\ [-1, 1] & \text{if } p = 0 \\ -1 & \text{if } p < 0 \end{cases} \quad (31)$$

Thus, we can write:

$$\partial f(x) = 4\text{sgn}(e_1^\top x)e_1 + 2\text{sgn}(e_2^\top x)e_2 \quad (32)$$

We should underline that $\partial f$ is rather a multifunction, that is, $\partial f(x)$ is a set for each $x$. For instance, for a point $x_0 = \alpha e_2$, with $\alpha > 0$, we have $\partial f(x_0) = 4[-1, 1]e_1 + 2e_2$, which represents the set pictured in Fig. 9.

![Figure 9: Image of $x_0 = \alpha e_2$, for $\alpha > 0$, by $\partial f$.](image)

In Fig. 10 we show how $\mathbb{R}^2$ is mapped by $\partial f$. One can easily verify that this graph is equal to the one of the multifunction $-\partial f$.

In this example, we can note that for any straight line $r$ crossing the origin in the dual plane $\partial f$, we have $r \cap \partial f(\mathbb{R}^2 \setminus \{0\}) \neq \emptyset$, that is, $r$ always crosses the boundary of the rectangle depicted in Fig. 10.
those activated subspaces. But if we generate an interesting solution, since it is not orthogonal to any of the regions I, II, III, IV (see Fig. 10) which does not represent the image of each point over the ordinates axes.

Since the hyperplane \( \Pi_c = \{ u \in \mathbb{R}^2 : c^\top u = 1 \} \) has a straight line \( r = \text{span}\{c\} \) as normal cone, then it will be possible to study minimality conditions of:

\[
\min_{c^\top u = 1} \|Y u\|_1,
\]

(33)

looking to the intersection \( (-\partial f(x)) \cap N_C(x) \neq \emptyset \) or \( (-\partial f(x)) \cap \text{span}\{c\} \neq \emptyset \) (see Fig. 11).

We can see that (for the choice \( c \) represented in Fig. 11) the solutions of (33) represent activated subspaces. In fact, the only problem is when we choose \( c \) such that \( \text{span}\{c\} \) intersects one of the vertices of the rectangle in the dual space \( \partial f \). In this case, the solution \( x^* \) would be in one of the regions I, II, III, IV (see Fig. 10) which does not represent an interesting solution, since it is not orthogonal to any of those activated subspaces. But if we generate \( c \) randomly (as in a roulette wheel), the probability that \( r \) crosses one of the vertices is zero, then all the solutions from (33) evaluated in this way will be always interesting.

It is important to note that the example given above for orthogonal subspaces is extensible to the oblique case. In this last case we obtain a general quadrilateral in the dual space instead of a rectangle. The trouble makers are still the vertices. This can also be generalized to \( \mathbb{R}^n \).

We present some results in Table 6 that confirm (latus sensus) our analysis.

<table>
<thead>
<tr>
<th>( \mathcal{L}_1 ) [%]</th>
<th>( \mathcal{L}_2 ) [%]</th>
<th>Rel. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>70</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 6: Results. The 100% in reliability means that the solutions obtained by solving the problem (33) (with an arbitrary choice for \( c \)) was always interesting.

5.2 Example: Outliers Effect

When there are outliers, the vertices of the rectangle in the dual space are widened as we now illustrate.

Let:

\[
Y = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}
\]

(34)

in which \( e = [1 \ 1] \) is clearly an outlier.

In this case we have \( \partial f(x) = 4\text{sgn}(e_1^\top x) + 2\text{sgn}(e_2^\top x) + \text{sgn}(e^\top x) \). The graph of \( \partial f \) is presented in Fig. 12.

We present some results in Table 6 that confirm (latus sensus) our analysis.
the associated minimizer of (33) is not an interesting point. In fact, such a solution would not be orthogonal to the activated subspaces, but it would be rather orthogonal to the outlier. Choosing $c$ arbitrarily, we have that the greater the number of outliers, the lower the probability to match interesting minimizers, since it will come up more "widened" zones. Actually, we can even estimate the likelihood of matching an interesting minimizer shooting $c$ randomly. We have:

$$p_a = 1 - \frac{2\theta}{360}.$$  

We can evaluate the value of $\theta$ by doing (see Fig. 12) $\theta = 90 - 45 - \arctan(\frac{2}{3}) = 33.69^\circ$, which results in $p_a = 81.28\%$.

We can proceed with a simulation of this example through the Sedumi and we can compare the theoretical and the experimental value of the reliability in matching interesting solutions:

$$\text{Reliability}_{\text{theoretical}} = 81.28\%$$
$$\text{Reliability}_{\text{experimental}} = 80.3\%$$  (35)

where it has been performed 10000 Monte-Carlos. For each Monte-Carlo an arbitrary unit-vector $c$ has been chosen.

6 Conclusion

We have presented two algorithms for segmenting arbitrary unions of linear subspaces: the MM-ELM routine in section 3 and the NS-ELM routine in section 4. Both approaches differ from most of current works by searching for activated subspaces by solving nicely structured optimization problems. The MM-ELM algorithm showed an interesting performance in dealing with the articulated case and even with real data. The main drawback is that the dimensions and number of subspaces are needed beforehand. In contrast, the NS-ELM algorithm does not need prior information regarding the number or dimensions of the subspaces. Following a recursive strategy it can peel all of the activated subspaces. It presented a reasonable performance in segmenting subspaces in the articulated case. However, this latter algorithm showed to be rather sensitive to noise and is not as computationally efficient as the MM-ELM algorithm in segmenting subspaces. We shall remark that the NS-ELM is initialized from 60 points whereas the MM-ELM is just initialized from one point following a suboptimal approach. Roughly speaking, the MM-ELM can in average segment two subspaces in about 3 seconds whereas the NS-ELM can spend up to 2 minutes in some cases.

We have presented also that the subspace segmentation problem can be partially solved through a linear program. We provided also an alternative geometrical approach of the problem through convex analysis.

As future work, the contraction operation from geometric algebra will be tested as cost function in the MM-ELM algorithm. Also, a richer characterization of the Grassmann manifold will be tried, that is, we will try to explore Riemannian properties of $\text{Proj}_{\text{n}_1}(\mathbb{R}_n)$ as a manifold. Some features from the physical problem will be taken into account in order to eliminate some physically non-feasible solutions from the model. A more mathematically rigorous analysis of the latter chapter is ongoing.

References