SPECTRAL UNMIXING USING DISTANCE GEOMETRY

Rob Heylen, Paul Scheunders

IBBT-Visionlab, University of Antwerp
Universiteitsplein 1, Building N
B-2610 Wilrijk (Antwerp), Belgium

ABSTRACT
In this paper, we present a new method for solving the spectral unmixing problem which uses only the spectral distances between the data points and the endmembers. This method is obtained by reformulating every step of the recently developed SPU algorithm entirely in distance geometry, yielding a recursive algorithm based on the geometrical properties of the spectral unmixing problem. The algorithm almost always minimizes the reconstruction error while obeying the constraints on the abundances, yielding results comparable to the fully-constrained least-squares solution. The performance of the algorithm is demonstrated on an artificial data set based on the USGS spectral library.

Index Terms—Hyperspectral imaging.

1. INTRODUCTION
Under the assumption of linear spectral mixing [1], the spectral unmixing problem involves the inversion of an over-determined linear system of equations, where one tries to minimize the reconstruction error, or Euclidean distance, between the data point and its reconstruction. Two constraints are usually imposed on the abundance coefficients due to their physical interpretation, positivity and sum-to-one, leading to a fully-constrained least-squares problem. Several techniques have been developed for solving this problem, based upon the introduction of a Lagrangian [2], quadratic programming [3], Bayesian methods [4], fuzzy sets [5]...

Recently, a new method has been developed for solving the spectral unmixing problem, the simplex-projection unmixing (SPU) algorithm [6]. This algorithm is based upon a geometrical interpretation of the problem, and recursively decreases the dimensionality by one in each iteration by identifying endmembers that have a zero abundance for a given pixel. The SPU algorithm requires the spectra of both the endmembers and the data points as input, as do all other spectral unmixing algorithms.

Under certain conditions however, one has only spectral distances between data points available. This is for instance the case if certain non-linear models or algorithms are used for unmixing, and the distances along the data manifold are the properties of interest, instead of the standard Euclidean coordinates and distances [7]. For such cases, it could be beneficial to have a spectral unmixing algorithm at our disposal that functions solely with spectral distances, and hence is expressed completely in distance geometry.

In this paper, we present such an algorithm, which requires only the distances between the data points and the endmembers as input. We first present the SPU algorithm, followed by the introduction of several properties from distance geometry needed for reformulating each step of the SPU algorithm in terms of inter-point distances. The performance of the resulting DSPU algorithm is compared to the original SPU algorithm, and the commonly used FCLSU algorithm.

2. UNMIXING VIA SIMPLEX PROJECTION
Consider $p$ endmembers $\{e_1, \ldots, e_p\}$ in $\mathbb{R}^d$. Every linear combination of these endmembers, obeying the positivity and sum-to-one constraint, lies in the simplex $S$, defined as

$$y \in S \iff \exists a_1, \ldots, a_p \in [0, 1] : \begin{cases} y = \sum_{i=1}^{p} a_i e_i \\ \sum_{i=1}^{p} a_i = 1 \end{cases}$$ (1)

The abundances $a = (a_1, \ldots, a_p)$ are homogeneous barycentric coordinates with respect to $\{e_1, \ldots, e_p\}$. Suppose one wants to unmix a given point $x$. In the linear mixing assumption, this corresponds to determining the set of abundances that minimizes the reconstruction error:

$$\min_a \|x - \sum_i a_i e_i\|_2$$ (2)

Geometrically, this corresponds to finding the abundances of the point inside the simplex $S$ with minimal distance to $x$. Hence, if we define the projection operator $P$ as

$$x' \in S : x' = P(x) \iff \forall y \in S : \|x - y\|_2 \geq \|x - x'\|_2$$ (3)

then the abundance coefficients of this projection will yield the solution to the unmixing problem, respecting both constraints on the abundances.

In [6], a method is proposed for efficiently executing this projection operation $P$, called the simplex-projection ummix-
Fig. 1. Two half-lines spanned by \((b, a)\) and \((b, c)\), and their angle bisector (dashed). The dotted line is perpendicular to the half-line \((b, a)\). In terms of closest-point projection onto these two lines, a point in region A will be projected onto \(b\), while a point in region B will be projected perpendicularly onto the half-line \((a, b)\). Hence the projection of a point in region A or B can always be written as a linear combination of \(a\) and \(b\), and \(c\) will never contribute. This property is independent of the angle between the two lines, and can be extended to higher dimensions.

Fig. 2. A generic 2-dimensional simplex, with the incenter and the three bissective cones indicated. A point in a cone \(Z_i\) will have a simplex projection with \(\hat{a}_i = 0\).

3. Determine the incenter \(c\), and identify the bissective cone \(Z_i\) that contains \(x\). Remove \(e_i\) from the set \(I\), and set \(a_i = 0\). Go to step 1.

4. Determine the abundances of \(x\) with respect to the endmembers in \(I\). This is now an exactly solvable linear system.

It has been shown [6] that the SPU algorithm has a high computational performance, and almost always minimizes the reconstruction error, the exception being some rare cases with simplices of very low relative volume (slivers) with highly obtuse angles. Furthermore, many steps in the algorithm depend only on the endmembers, allowing for an efficient implementation if more than one data point needs to be projected, contrary to many alternative algorithms that only treat point per point.

3. INCORPORATING DISTANCE GEOMETRY.

In distance geometry, one describes and solves a geometrical problem entirely in terms of (Euclidean) distances between the different constituents. Here, we want to reformulate the SPU algorithm within the framework of distance geometry. We therefore assume that only the mutual distances between the endmembers are known, and the distances from the data point \(x\) to the endmembers. All properties should be expressed in terms of distances to the endmembers.

The different steps of the algorithm that need to be reformulated are orthogonal projection onto a plane, determining the abundance coefficients of a point, determining the incenter, and identifying the bissective cone that contains a given point. We assume \(p > 2\), since \(p = 1\) and \(p = 2\) are trivial.

3.1. The Cayley-Menger determinant and orthogonal projection

An important property that we will use several times is the Cayley-Menger determinant, describing the volume of a sim-
plex in terms of the inter-point distances. Let \( d_{ij} \) be the Euclidean distance between points \( e_i \) and \( e_j \). The volume \( V \) of a simplex spanned by \( e_1, \ldots, e_p \) is then given by:

\[
(-1)^p 2^{p-1} ((p-1)!)^2 V^2 = \begin{vmatrix} D_{1\ldots p} & 1 \\ 1 & 0 \end{vmatrix} \quad (5)
\]

with \( D_{1\ldots p} = [d_{ij}]_{i,j=1.2\ldots p} \) the squared distance matrix.

From this equation, one can derive that the orthogonal distance from \( e_1 \) to the plane spanned by vertices \( (e_2, \ldots, e_p) \) is given by

\[
d^2(e_1; e_2, \ldots, e_p) = \frac{d_1^2 C_{2\ldots p}^{-1} d_1}{2} \quad (7)
\]

with \( d_1 = (d_{12}^2, \ldots, d_{1p}^2, 1) \). This equation allows us to find the orthogonal projection \( x_{\perp} \) of a point \( x \) onto a plane spanned by vertices \( (e_1, \ldots, e_p) \) by using Pythagoras’ rule to find the new distances of \( x_{\perp} \) to the vertices:

\[
d^2(x_{\perp}, e_i) = d^2(x, e_i) - d^2(x; e_1, \ldots, e_p) \quad (8)
\]

### 3.2. The incenter

The coordinates of the incenter \( c \) can be found via the \((p-2)\)-dimensional volumes of the \( p \) faces of the simplex. Let \( V_i \) be the volume of the sub-simplex spanned by \( e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_p \). Then we have

\[
a^c_i = \frac{V_i}{\sum_{i=1}^{p} V_i} \quad (9)
\]

yielding a vector \( \alpha^c \) containing the barycentric coordinates of the incenter \( c \). The volumes can be found with (6).

Once the barycentric coordinates of the incenter are known, we can calculate the distances from the incenter to the endmembers with the following result from multidimensional scaling. The squared distance between two points \( x \) and \( y \), with respective abundances \( a_x \) and \( a_y \), is given by:

\[
d^2(x, y) = (a_x - a_y)^T \left( -\frac{1}{2} J D J \right) (a_x - a_y) \quad (10)
\]

where the elements of the \( p \times p \) centering matrix \( J \) are given by \( J_{ij} = \delta_{ij} - \frac{1}{p} \).

### 3.3. Identification of bissective cone

In order to identify the bissective cone that contains a given point, we must be capable to complete a distance matrix with one distance missing. Consider \( p \) points \( e_1, \ldots, e_p \) in \( \mathbb{R}^p \), with squared distance matrix \( D \), and two points \( x \) and \( y \) of which we know only the squared distances to the points \( \{e_i\} \): \( d^2 = (d^2(x, e_1), \ldots, d^2(x, e_p)) \), \( d^2 = (d^2(y, e_1), \ldots, d^2(y, e_p)) \), and \( D = [d^2(e_i, e_j)]_{i,j=1,2\ldots p} \).

The distance between \( x \) and \( y \) is \( d^2(x, y) = x \), and considered unknown. Because we have \( p + 2 \) points in total in a \( p \)-dimensional space, this distance can take on only two values, corresponding to the case where \( x \) and \( y \) are on the same side of the hyperplane through \( \{e_i\} \) or on opposite sides. If one has more than \( p + 2 \) points, the following reasoning can be applied to any subset of \( p + 2 \) points.

The \( p + 2 \) points form a \((p+1)\)-dimensional simplex of volume zero, since all points are embedded in \( \mathbb{R}^p \). Via the Cayley-Menger determinant (6), we find

\[
\begin{vmatrix}
0 & 1 & 1 & 1 & \cdots & 1 \\
1 & 0 & x & d^2_1 & \cdots & d^2_p \\
1 & x & 0 & d^2_1 & \cdots & d^2_p \\
1 & d^2_1 & d^2_1 & D_{11} & \cdots & D_{1p} \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
1 & d^2_p & d^2_p & D_{p1} & \cdots & D_{pp}
\end{vmatrix} = 0 \quad (11)
\]

By using the following determinant property

\[
\begin{vmatrix}
X \\
Y \\
D
\end{vmatrix} = |D| \begin{vmatrix} X - Y D^{-1} Y^T \\
\end{vmatrix} \quad (12)
\]

we find after some algebra a second-order equation in \( x \), yielding the two possibilities for the unknown distance between \( x \) and \( y \): the smaller distance corresponds to \( x \) and \( y \) being on the same side of the plane through the endmembers, while the larger distance corresponds to them being on opposite sides.

Now, to identify the bissective cone that contains a given point \( x \), we use the following reasoning: A cone \( Z_i \) is determined by the incenter \( c \) and the \( p - 1 \) vertices \( \{e_j\}_{j \neq i} = \{e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_p\} \). We can then define \( p - 1 \) subcones \( Z_{ij} \), defined by the incenter \( c \) and the vertices in \( Z_i \) except \( e_j \). The point \( x \) now lies in the cone \( Z_i \) if and only if for all \( j \), \( x \) and \( e_j \) lie on the same side of the plane through the points of sub-cone \( Z_{ij} \). This can be checked by calculating the two possible distances between \( x \) and \( e_j \) via distance matrix completion, and comparing the known distance with the two possibilities.

### 3.4. Determination of abundance coefficients.

The last operations we will need is determining whether a point lies inside a simplex or not, and finding its abundance coefficients. For the former, we can use the observation that a point lies inside the simplex if and only if for all \( i \), \( x \) and \( e_i \) lie on the same side of the plane through \( \{e_j\}_{j \neq i} \). To determine the abundance coefficients of a point \( x \) lying within a simplex, we can use the equivalence between abundances, homogeneous barycentric coordinates, and areal coordinates:

\[
a_i = \frac{V(e_1, e_{i-1}, x, e_{i+1}, \ldots, e_p)}{V(e_1, \ldots, e_p)} \quad (13)
\]

or abundance \( a_i \) is the normalized volume of the simplex obtained by replacing the \( i \)-th vertex by \( x \).
4. RESULTS

For demonstrating the algorithm, we have created an artificial data set in the following way: Randomly select $p$ spectra from the USGS spectral database as endmembers, and interpolate them at 50 evenly spaced wavelengths in the range 1.97-2.48 $\mu$m. Generate an abundance matrix $A$ of $N$ randomly and uniformly selected abundance vectors respecting the positivity and sum-to-one constraint, and a noise matrix with a chosen signal-to-noise (SNR) level [8]. Generate $N$ data points from the endmembers and abundances with the linear mixing equation, and add the random noise.

As a first experiment, we unmixed this data set with respect to the known endmembers with three different algorithms: The FCLSU algorithm presented in [2], the original SPU algorithm from [6], and the DSPU algorithm presented in this paper. We chose $p = 8, N = 10^5, SNR = 10$, and calculated the mean, median and maximum of the absolute differences between the $8 \cdot 10^5$ abundance coefficients retrieved with the three methods. See table 1 for a typical result.

As can be seen, the median and mean errors are very small, and can be considered numerical noise. The maximum errors however can be larger, which after calculation of the reconstruction errors indicates that there are some pixels where the FCLSU solution provides a slightly better result. The main factors causing this deviance between the (D)SPU and the FCLSU methods are the sensitivity to numerical noise and correlations between spectral bands and endmembers.

Next, we have examined the performance of the algorithm by measuring the average runtime as a function of the number of pixels $N$ in Fig. 3. This figure indicates that both the SPU and DSPU algorithm have a significantly lower runtime than the FCLSU algorithm. The DSPU algorithm is slower than the SPU algorithm for lower $N$, but the runtimes converge for higher $N$.

5. CONCLUSIONS

We have introduced a new method for supervised spectral unmixing that uses only the spectral distances between the data points and the endmembers as input. When tested on a simulated data set, the results obtained with this new method are comparable to those obtained with the known fully-constrained least-squares unmixing (FCLSU) technique, the only differences caused by the larger numerical sensitivity of the proposed method. The new algorithm is shown to be an order of magnitude faster than the traditional FCLSU algorithm.

6. REFERENCES