Simplex Volume Analysis Based On Triangular Factorization: A framework for hyperspectral Unmixing

- Wei Xia, Bin Wang, Liming Zhang, and Qiyong Lu
- Dept. of Electronic Engineering
- Fudan University, China
1. Introduction
2. The Proposed Method
   2.1 Endmember extraction
   2.2 Abundance Estimation
3. Evaluation with Experiments
   3.1 Synthetic data
   3.2 Real hyperspectral data
4. Conclusion
1. Introduction

2. The Proposed Method
   2.1 Endmember extraction
   2.2 Abundance Estimation

3. Evaluation with Experiments
   3.1 Synthetic data
   3.2 Real hyperspectral data

4. Conclusion
Linear Mixture Model (LMM)

The observation of a pixel

\[ x = As + e \]

Abundance fractions

endmember spectra

\[ x \in \mathbb{R}^{L \times 1}, \]

\[ A \in \mathbb{R}^{L \times P}, \]

\[ s \in \mathbb{R}^{P \times N} \]

\[ s_i \geq 0, \quad (i = 1, 2, \ldots, P). \]

\[ \sum_{i=1}^{P} s_i = 1 \]
Different methods under the LMM

Geometrical category
- PPI
- N-FINDR
- VCA
- SGA
- OBA
- ...

Statistical category
- ICA-based
- NMF-based
- ...

A Simplex of P-vertices is defined by

\[
\begin{align*}
\mathbf{x} &= s_1 \mathbf{e}_0 + s_1 \mathbf{e}_1 + \ldots + s_{P-1} \mathbf{e}_{P-1} \\
\text{subject to} & \quad s_i > 0, \quad \sum_{i=1}^{P} s_i = 1
\end{align*}
\]
Simplex Volume Analysis (2/2)

Related work *

- The observation pixels forms a simplex whose vertices correspond to the endmembers
- Find the vertices by searching for the pixels which can form the largest volume of the simplex

Volume formula

\[ V = \frac{1}{(P-1)!} \det \left( \begin{bmatrix} 1^T \\ E \end{bmatrix} \right) \]

\[ E = [e_0, e_1, \ldots, e_{P-1}] \]

Disadvantages

- large computing cost caused by calculating volume, hard to be used for Real-time application
- Require dimensionality reduction (DR), loss of possible information

---


1. Introduction

2. The Proposed Method
   2.1 Endmember extraction
   2.2 Abundance Estimation

3. Evaluation with Experiments
   3.1 Synthetic data
   3.2 Real hyperspectral data

4. Conclusion
The Proposed Method base on Triangular Factorization (TF)

Proposed Method

- Estimate A: SVATF (Simplex Volume Analysis based on Triangular Factorization)
  - Basic idea: simplify the computation by utilizing TF
- Estimate S: AQTF (Abundance Quantification based on Triangular Factorization)
  - Basic idea: rectify the possible errors by using TF

Hyperspectral Unmixing $X = AS$

<table>
<thead>
<tr>
<th>Endmember</th>
<th>Abundance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extraction---A</td>
<td>Estimation---S</td>
</tr>
</tbody>
</table>
Proposed Endmember Extraction Framework

SVATF (Simplex Volume Analysis based on Triangular Factorization)

\[ \mathbf{A} = [\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{p-1}] \]

\[ \tilde{\mathbf{A}} = [\mathbf{e}_1 - \mathbf{e}_0, \mathbf{e}_2 - \mathbf{e}_0, \ldots, \mathbf{e}_{p-1} - \mathbf{e}_0] \]

• Simplex Volume

\[ V = \frac{1}{(P-1)!} \left| \det(Z) \right|^\frac{1}{2} \]

\[ Z = \tilde{A}^T \tilde{A} = \begin{bmatrix}
\| \mathbf{a}_1 \|^2 & \mathbf{a}_1 \cdot \mathbf{a}_2 & \ldots & \mathbf{a}_1 \cdot \mathbf{a}_{P-1} \\
\mathbf{a}_2 \cdot \mathbf{a}_1 & \| \mathbf{a}_2 \|^2 & \ldots & \mathbf{a}_2 \cdot \mathbf{a}_{P-1} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{a}_{P-1} \cdot \mathbf{a}_1 & \mathbf{a}_{P-1} \cdot \mathbf{a}_2 & \ldots & \| \mathbf{a}_{P-1} \|^2
\end{bmatrix} , \text{ (where } \mathbf{a}_i = \mathbf{e}_i - \mathbf{e}_0 \text{)}

• \( Z \) is a positive definite symmetric matrix, which can be decomposed by Cholesky Factorization

\[ Z = LL^T \]
Develop by Cholesky Factorization (2/5)

- Update the Simplex Volume

\[ V = \frac{1}{(P-1)!} \left| \det(Z) \right|^{\frac{1}{2}} \]

\[ = \frac{1}{(P-1)!} \left| \det(LL^T) \right|^{\frac{1}{2}} = \frac{1}{(P-1)!} |l_{11}| |l_{22}| \cdots |l_{(P-1)(P-1)}| \]

- Calculating the simplex volume

\[ \text{Perform the Cholesky factorization} \]

- Maximize the volume \( V \)

\[ \text{maximizing diagonal element } |l_{i,i}|, \ (i = 1, 2, ..., P-1) \]
How does SVATF run?

\[ Z = LL^T \]

\[ l_{i,i} = \left( z_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2 \right)^{1/2} \]

\[ l_{i,j} = \frac{z_{i,j} - \sum_{k=1}^{i-1} l_{i,k} l_{j,k}}{l_{j,j}}, \quad \text{for } i > j. \]

- Find the endmember, i.e., search for the pixel which can maximize \( l_{i,i} \).

1. search for \( \mathbf{e}_1 \) \( l_{1,1} = \sqrt{z_{1,1}} \) \( \ldots \) \( i=1 \)

2. search for \( \mathbf{e}_2 \) \( l_{2,2} = \sqrt{z_{2,2} - l_{2,1}^2} \) \( \ldots \) \( i=2 \)

3. search for \( \mathbf{e}_3 \) \( l_{3,3} = \sqrt{z_{3,3} - l_{3,1}^2 - l_{3,2}^2} \) \( \ldots \) \( i=3 \)

Easy to realize:
Calculate Cholesky Factorization for \( N \) times
to find all the endmembers (\( N \) is the number of pixels).
The benefit of using Cholesky Factorization

• Simplify the searching process

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>The number of calculated Determinants*</th>
<th>The matrix in calculated Determinant*</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-FINDR (after DR)</td>
<td>$NP$</td>
<td>$P \times P$ size matrix</td>
</tr>
<tr>
<td>SGA (after DR)</td>
<td>$Nn$ $(n$ starting from 2 to $P)$</td>
<td>$n \times n$ size matrix</td>
</tr>
<tr>
<td>SVATF (With/Without DR)</td>
<td>$N$</td>
<td>$(P-1) \times (P-1)$ size matrix</td>
</tr>
</tbody>
</table>

• SVATF calculate the determinants on a smaller matrix using fewer number

SVATF can perform faster with/without DR

---

Develop by Cholesky Factorization (5/5)

Given the observation matrix $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{L \times N}$, $P$: endmember number

**Step 1 Initialization**
- Search for $e_o = \arg \max_{x_n} (\| x_n \|)$, $(n = 1, 2, ..., N)$
- Search for $e_1 = \arg \max_{x_n} (\| x_n \|)$, $(\tilde{x}_n = x_n - e_o)$
- Set $t=1$, $\gamma_n^1 = \| \tilde{x}_n \|$, $id(1) = \arg \max_n (\gamma_n^1)$

**Step 2 Start iteration**
- Calculate $\eta_n^t = (\tilde{x}_n \cdot a_t - \sum_{k=1}^{t-1} \eta_n^k \eta_{id(t)}^k) / \gamma_n^t$, $\gamma_n^{t+1} = \sqrt{(\gamma_n^t)^2 - (\eta_n^t)^2}$, $a_t = e_t - e_o$
- Search for $e_{t+1} = \arg \max_{x_n} (\gamma_n^{t+1})$, $id(t+1) = \arg \max_n (\gamma_n^{t+1})$
- $t = t+1$. Go back to Step 2 if $t < P-1$

**Step 3 Output**
- results $A = [e_0, e_1, ..., e_{P-1}]$
### Computational complexity among N-FINDR, VCA, SGA, OBA, and SVATF

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Numbers of flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-FINDR</td>
<td>$P^{\eta+1}N$</td>
</tr>
<tr>
<td>SGA</td>
<td>$(\sum_{k=2}^{p} k^\eta)N$</td>
</tr>
<tr>
<td>VCA</td>
<td>$2P^2N$ .... after dimensionality reduction $2PLN$ .... without dimensionality reduction</td>
</tr>
<tr>
<td>OBA*</td>
<td>$N(3P^2 - 4P + 1) + P - 1$ .... after dimensionality reduction $N(P - 1 + 3PL - 2L) + L$ .... without dimensionality reduction</td>
</tr>
<tr>
<td>SVATF</td>
<td>$0.5N(3P^2 - P - 4)$ .... after dimensionality reduction $0.5N(P^2 + 2PL + P - 4)$ .... without dimensionality reduction</td>
</tr>
</tbody>
</table>

The numbers of flops in various endmembers

The flops of dimensionality reduction: $> 2NL^2$

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$L$</th>
<th>$P$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>3, 4,…, 50</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

The number of endmembers

The number of floating operations

- VCA
- SGA
- NFINDR
- OBA
- SVATF

The number of endmembers vs. the number of floating operations graph.
The numbers of flops in various pixels

<table>
<thead>
<tr>
<th>Parameters</th>
<th>( L )</th>
<th>( P )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>10</td>
<td>100, 1000, …, 1e+8</td>
</tr>
</tbody>
</table>

The number of floating operations vs. the number of pixels for different algorithms (VCA, SGA, NFINDR, OBA, SVATF).
The numbers of flops in various bands

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
</tr>
<tr>
<td>100, 200, ..., 800</td>
</tr>
</tbody>
</table>

![Graph showing the number of floating operations for different parameters and algorithms.](image)
1. Introduction

2. The Proposed Method
   2.1 Endmember extraction
   2.2 Abundance Estimation

3. Evaluation with Experiments
   3.1 Synthetic data
   3.2 Real hyperspectral data

4. Conclusion
Abundance Quantification based on TF

- Known the endmembers, the abundances can be given as
  \[ X = AS \quad \longrightarrow \quad S = \text{inv}(A)X \]

- Transform into
  \[ x = QRs \quad \quad \quad Q^T x = Rs \]

- Estimate the abundance \( s_i \), \( i = 1, 2, \ldots, P \) by solving linear simultaneous equation
• Resulting formula

\[
\begin{align*}
    s_p &= \frac{b_p}{r_{pp}} \\
    s_{p-1} &= \frac{b_{p-1}}{r_{p-1,p-1}} \\
    \vdots \\
    s_1 &= \frac{b_1 - \sum_{i=2}^{p} r_{i} s_i}{r_{11}}
\end{align*}
\]

where

\[
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_p
\end{bmatrix} = Q^T 
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_L
\end{bmatrix}
\]
Similarly, obtain $A$ when $S$ is known

\[ X = AS \]

\[ A = QR \]

\[ Q^T x = Rs \]

\[ S = \text{inv}(R)Q^T X \]

\[ X^T = S^T A^T \]

\[ S^T = Q_S R_S \]

\[ Q_S^T X^T = R_S A^T \]

\[ A = \left( \text{inv}(R_S) Q_S^T X^T \right)^T \]
\[ A = \lambda \left( \text{inv}(R_S) \cdot Q^T_s \cdot X^T_t \right)^T + (1 - \lambda) A \]
1. Introduction

2. The Proposed Method
   2.1 Endmember extraction
   2.2 Abundance Estimation

3. Evaluation with Experiments
   3.1 Synthetic data
   3.2 Real hyperspectral data

4. Conclusion
Experiments

Algorithms

• **N-FINDR** (M. E. Winter 1999) ***

• **SGA** (Chang, Wu, Liu, & Ouyang, 2006) **

• **VCA** (Nascimento & Bioucas-Dias, 2005) *

---


Criteria

- **SAD**
  
  $SAD_i = \arccos \frac{a_i^T \hat{a}_i}{\|a_i\| \cdot \|\hat{a}_i\|}$

  - $a_i \in \mathbb{R}^{L \times L}$: The spectral of the $i$th endmember
  - $\hat{a}_i \in \mathbb{R}^{L \times P}$: The estimated spectral of the $i$th endmember

- **RMSE**
  
  $RMSE_i = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (y_{ij} - s_{ij})^2}$

  - $s_{ij}$: The abundance of the $i$th endmember according to the $j$th pixel
  - $y_{ij}$: The estimated abundance of the $i$th endmember according to the $j$th pixel
Synthetic Data (1/5)


The abundances fractions are subject to Dirichlet distribution.
Synthetic Data (2/5)

Results of the algorithms with Different **image sizes**

<table>
<thead>
<tr>
<th>Image Size</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>100×100</td>
<td>10^-2</td>
</tr>
<tr>
<td>200×200</td>
<td>10^-1</td>
</tr>
<tr>
<td>300×300</td>
<td>10^0</td>
</tr>
<tr>
<td>400×400</td>
<td>10^1</td>
</tr>
<tr>
<td>500×500</td>
<td>10^2</td>
</tr>
<tr>
<td>600×600</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU</th>
<th>memory</th>
<th>OS</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel(R) Xeon CPU</td>
<td>48 GBytes</td>
<td>64-bit Window7</td>
<td>Matlab 2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L)</td>
</tr>
<tr>
<td>224</td>
</tr>
</tbody>
</table>

SVATF without DR
Other Methods: use DR
Results of the algorithms with different mixing degrees
Results of the algorithms with Different noise levels
The effectiveness of AQTF with different mixing degrees
Real Data-Cuprite(1/3)

- Cuprite dataset *
  - 224 bands
  - spectral resolution 10nm
  - captured by AVIRIS in June 1997
Estimated abundance maps

- (a) Muscovite #1, (b) Desert_Varnish, (c) Alunite, (d) Kaolinite #1, (e) Montmorillonite #1, (f) Kaolinite #2, (g) Buddingtonite, (h) Jarosite, (i) Nontronite, (j) Chalcadony, (k) Kaolinite #3, (l) Muscovite #2, (m) Sphene, (n) Montmorillonite #2.
The Spectra obtained by SVATF

(a) Muscovite #1, (b) Desert Varnish, (c) Alunite, (d) Kaolinite #1, (e) Montmorillonite, (f) Jarosite, (g) Buddingtonite, (h) Kaolinite #2, (i) Nontronite, (j) Chalcadony, (k) Kaolinite #3, (l) Muscovite #2, (M) Sphene, (n) Montmorillonite #2

- Solid line: Reference,
- Dashed line: Estimated result
The comparison of SAD

<table>
<thead>
<tr>
<th>Index</th>
<th>Reference Spectra</th>
<th>N-FINDR</th>
<th>SGA</th>
<th>VCA</th>
<th>SVATF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Muscovite GDS108</td>
<td>0.0900</td>
<td>0.0724</td>
<td>0.1631</td>
<td>0.0801</td>
</tr>
<tr>
<td>2</td>
<td>Desert_Varnish GDS141</td>
<td>0.2252</td>
<td>0.1599</td>
<td>0.2454</td>
<td>0.1595</td>
</tr>
<tr>
<td>3</td>
<td>Alunite GDS82 Na82</td>
<td>0.0690</td>
<td>0.0690</td>
<td>0.2172</td>
<td>0.0714</td>
</tr>
<tr>
<td>4</td>
<td>Kaolinite KGa-2</td>
<td>0.2574</td>
<td>0.2201</td>
<td>0.2201</td>
<td>0.2586</td>
</tr>
<tr>
<td>5</td>
<td>Montmorillonite+Ilili CM37</td>
<td>0.1519</td>
<td>0.1259</td>
<td>0.0544</td>
<td>0.0501</td>
</tr>
<tr>
<td>6</td>
<td>Kaolinite CM7</td>
<td>0.2530</td>
<td>0.2550</td>
<td>0.1769</td>
<td>0.0814</td>
</tr>
<tr>
<td>7</td>
<td>Buddingtonite GDS85 D-206</td>
<td>0.0761</td>
<td>0.1598</td>
<td>0.1053</td>
<td>0.0674</td>
</tr>
<tr>
<td>8</td>
<td>Jarosite GDS98</td>
<td>0.2812</td>
<td>0.2113</td>
<td>0.2368</td>
<td>0.2163</td>
</tr>
<tr>
<td>9</td>
<td>Nontronite NG-1.a</td>
<td>0.0717</td>
<td>0.1374</td>
<td>0.0741</td>
<td>0.0682</td>
</tr>
<tr>
<td>10</td>
<td>Chaledony CU91</td>
<td>0.1241</td>
<td>0.1666</td>
<td>0.1317</td>
<td>0.0727</td>
</tr>
<tr>
<td>11</td>
<td>Kaolinite GDS11 &lt;63um</td>
<td>0.1870</td>
<td>0.1896</td>
<td>0.2376</td>
<td>0.1752</td>
</tr>
<tr>
<td>12</td>
<td>Muscovite IL107</td>
<td>0.1019</td>
<td>0.0995</td>
<td>0.0888</td>
<td>0.0801</td>
</tr>
<tr>
<td>13</td>
<td>Sphene HS189.3B</td>
<td>0.2128</td>
<td>0.1502</td>
<td>0.0970</td>
<td>0.0677</td>
</tr>
<tr>
<td>14</td>
<td>Montmorillonite Sca2b</td>
<td>0.1298</td>
<td>0.1206</td>
<td>0.1103</td>
<td>0.1674</td>
</tr>
<tr>
<td></td>
<td>sum SAD values</td>
<td>2.2311</td>
<td>2.1749</td>
<td>2.1587</td>
<td>1.6161</td>
</tr>
</tbody>
</table>
### Computing time for the Cuprite dataset

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>NFINDR-FCLS</th>
<th>SGA-FCLS</th>
<th>VCA-FCLS</th>
<th>SVATF-AQTF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NFINDR</td>
<td>FCLS</td>
<td>SGA</td>
<td>FCLS</td>
</tr>
<tr>
<td>Time</td>
<td>28.27780</td>
<td>16.63869</td>
<td>3.13832</td>
<td>16.38314</td>
</tr>
</tbody>
</table>

### The computer environment

<table>
<thead>
<tr>
<th>CPU</th>
<th>memory</th>
<th>OS</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel(R) Xeon CPU</td>
<td>48 GBytes</td>
<td>64-bit Window7</td>
<td>Matlab 2010</td>
</tr>
</tbody>
</table>
Contents

1. Introduction

2. The Proposed Method

   2.1 Endmember extraction

   2.2 Abundance Estimation

3. Evaluation with Experiments

   3.1 Synthetic data

   3.2 Real hyperspectral data

4. Conclusion
Conclusion

• Proposed a new method based on triangular factorization for the simplex analysis of hyperspectral unmixing.

• A framework including a group of algorithms.

• Dimensionality reduction (DR) is optional.

• Efficiency and accuracy. Both the theoretical analysis and experimental results show that the proposed methods can perform faster than the state-of-the-art methods, with precise results.

  Should be very useful for Real-time application.

• Steady. always outputs the same results in the same sequence when being applied to a certain dataset.
THANK YOU