CS 754 - Advanced Image Processing
Assignment 4 - Report

Shaan ul Haque - 180070053
Mantri Krishna Sri Ipsit - 180070032

April 3, 2021

2

The algorithm other than OMP and MP presented is called as **Look Ahead Orthogonal Matching Pursuit**. The standard OMP serially adds one component to the support-set that will irreversibly remain in the final estimate, in each iteration. The current choice of the component chosen is independent of the final result. This shortcoming can be overcome using a *look ahead* strategy. In this method, an element is chosen by evaluating its effect on the final performance, in the sense of minimizing the fitting residual at the end of all future iterations. In the current iteration, several potential elements are chosen by the matched filter MF. Each of the potential elements are tested independently, and then one element among them is selected and inducted to the support-set. Among the potential elements, the element that will be selected provides the minimum fitting residual after executing all the future iterations (look ahead strategy). This method is referred to as LAOMP. To develop the LAOMP algorithm, the following algorithm is helpful:

Some notations that were followed:

$$\begin{aligned}
\texttt{resid}(\boldsymbol{y}, \boldsymbol{A}_{\mathcal{T}}) &\triangleq \boldsymbol{y} - \boldsymbol{y}_p, \quad \text{where} \quad \boldsymbol{y}_p = \boldsymbol{A}_{\mathcal{T}} \boldsymbol{A}_{\mathcal{T}}^{\dagger} \boldsymbol{y} \\
\texttt{supp}(\boldsymbol{x}, k) &\triangleq \{\text{the set of indices corresponding to the } k \text{ largest components of } \boldsymbol{x}\} \\
\texttt{supp}(\boldsymbol{x}) &\triangleq \{\text{the set of indices corresponding to all the non-zero components of } \boldsymbol{x}\}
\end{aligned}$$

Please turn over

```
Algorithm 1: look_ahead_resid: Look Ahead Residual
     Input          : $\boldsymbol{A}, \boldsymbol{y}, T$, previous support-set $\mathcal{T}_{\text{old}}$, new index $\tau$
     Assumption : $i \notin \mathcal{T}_{\text{old}}$, $\hat{\boldsymbol{x}} \in \mathbb{R}^N$
     Initialization:
1    $k = |\mathcal{T}_{\text{old}}| + 1$;
2    $\mathcal{T}_k \leftarrow \mathcal{T}_{\text{old}} \cup \tau$;
3    $\mathbf{r}_k \leftarrow \texttt{resid}(\mathbf{y}, \mathbf{A}_{\mathcal{T}_k})$;
     Iteration      :
4    repeat
5        $k \leftarrow k + 1$;
6        $\tau_k \leftarrow \texttt{supp}(\mathbf{A}^{\intercal}\mathbf{r}_{k-1}, 1)$;
7        $\mathcal{T}_k \leftarrow \mathcal{T}_{k-1} \cup \tau_k$;
8        $\mathbf{r}_k \leftarrow \texttt{resid}(\mathbf{y}, \mathbf{A}_{\mathcal{T}_k})$;
9    until $k{=}T$;
     Output         : $\mathbf{r} \leftarrow \mathbf{r}_k$
```

The above algorithm is the look ahead part of LAOMP. Given the sparsity level T , an intermediate support set $\mathcal{T}$ and a new choice of an atom index τ, the algorithm usually finds a T-element support set at the end and returns the final least squares (LS) residual.
The LAOMP algorithm is based on OMP to construct the support set progressively. In LAOMP, for the choice of a new atom, we take into account its effect on the final performance in the sense of minimizing the residual. We fix an integer parameter λ which is referred to as the look ahead parameter. For the kth iteration, the λ best choices of new atoms are found using MF (Step 3). Then, for each choice of λ atoms, we allow the algorithm to execute until it selects T atoms (Steps 4 and 5. From the λ atoms we choose the one that results in the minimum norm of fitting residual (Step 9). The chosen atom is added to the intermediate support set $\mathcal{T}_{k-1}$ to form the new support set $\mathcal{T}_k$ of cardinality k. The rest of the algorithm remains same as standard OMP. We note that the algorithm must stop when the number of iterations exceeds T . Like most GP algorithms, we assume that the a-priori knowledge of the sparsity level T is available. Also note that, as the value of the look ahead parameter λ increases, better performance can be expected, but at the expense of higher complexity. For $\lambda = 1$, LAOMP will provide the same result as OMP.

Please turn over

Algorithm 2: Look ahead orthogonal matching pursuit (LAOMP)

```
Input          : $\boldsymbol{A}, \boldsymbol{y}, T$, look ahead parameter $\lambda \leq T$
Define         : $\boldsymbol{n} = [n_1, n_2, \ldots, n_\lambda]^\intercal$ and $\mathcal{T}_\Lambda = [\mathcal{T}_\Lambda(1)], \mathcal{T}_\Lambda(2), \ldots, \mathcal{T}_\Lambda(L)]$
Initialization:
$k \leftarrow 0, l \leftarrow 0$;
$\boldsymbol{r}_k \leftarrow \boldsymbol{y}, \hat{\boldsymbol{x}}_k \leftarrow \mathbf{0}, \mathcal{T}_k \leftarrow \emptyset$;
Iteration      :
repeat
    $k \leftarrow k + 1$;
    $\mathcal{T}_\Lambda \leftarrow \texttt{supp}(\boldsymbol{A}^\intercal \boldsymbol{r}_{k-1}, \lambda)$;
    for $l = 1$ to $\lambda$ do
        $\boldsymbol{rr} \leftarrow \texttt{look\_ahead\_resid}\big(\boldsymbol{y}, \boldsymbol{A}, T, \mathcal{T}_{k-1}, \mathcal{T}_\Lambda(l)\big)$;
        $n_l \leftarrow \|\boldsymbol{rr}\|_2$
    end for
    $l \leftarrow$ Smallest component in $\mathbf{n}$;
    $\tau_k \leftarrow \mathcal{T}_\Lambda(l)$;
    $\mathcal{T}_k \leftarrow \mathcal{T}_{k-1} \cup \tau_k$;
    $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ such that $\hat{\boldsymbol{x}}_{\mathcal{T}_k} = \boldsymbol{A}^\dagger_{\mathcal{T}_k} \boldsymbol{y}$ and $\hat{\boldsymbol{x}}_{\mathcal{T}_k^c} = \mathbf{0}$;
    $\boldsymbol{r}_k \leftarrow \texttt{resid}(\boldsymbol{y}, \boldsymbol{A}_{\mathcal{T}_k})$
until $k = T$;
Output         : $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ such that $\hat{\boldsymbol{x}}_{\mathcal{T}_k} = \boldsymbol{A}^\dagger_{\mathcal{T}_k} \boldsymbol{y}$ and $\hat{\boldsymbol{x}}_{\mathcal{T}_k^c} = \mathbf{0}$
```

As this algorithm is based on OMP, the following theorem which presents the performance bounds is also valid for LAOMP. (There is not seperate theorem stating performance bounds was given for LAOMP)

Theorem 1 *Suppose that $\boldsymbol{A}$ satisfies the restricted isometry property (RIP) of order $T+1$ with isometry constant $\delta_{T+1} < \frac{1}{3\sqrt{T}}$. Then, for any $\boldsymbol{x} \in \mathbb{R}^N$ with the number of non-zeros less than or equal to s, LAOMP (OMP) will recover $\boldsymbol{x}$ exactly from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ in T iterations.*

Theorem 2 (Restricted Isometry Property) *A matrix A is said to obey the restricted isometry property of order N if for any N-sparse vector x*

$$(1 - \delta_N) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_N) \|x\|_2^2$$

where $0 \leq \delta_N \leq 1$ is the smallest positive number such that the above relation holds and it is called "restricted isometry constant" or "isometry constant".

4

4.1

$$J(\boldsymbol{A_r}) = \|\boldsymbol{A} - \boldsymbol{A_r}\|_F^2$$

Solving this optimization problem is nothing but finding the best r-rank approximation of the matrix $\boldsymbol{A}$. As we have seen in the lecture, this is done by using the top r-largest singular values and reconstructing $\boldsymbol{A_r}$. This is also known as **Eckart–Young–Mirsky theorem**.

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{USV}^\intercal \\ \implies \boldsymbol{A_r} &= \boldsymbol{U}(:,1:r)\boldsymbol{S}(1:r,1:r)\Big(\boldsymbol{V}(:,1:k)\Big)^\intercal \end{aligned}$$

This low rank matrix approximation can be used in **Texture Completion**. Generally the texture or any repeating pattern is captured by the low rank approximation. This optimization problem can be used to remove occlusions either from a photo or a video and do a robust repair, which is an application of **Robust PCA**.

4.2

$$J(\boldsymbol{R}) = \|\boldsymbol{A} - \boldsymbol{RB}\|_F^2$$

where $\boldsymbol{R}$ is an orthonormal matrix i.e, $\boldsymbol{RR}^\intercal = \boldsymbol{R}^\intercal\boldsymbol{R} = \boldsymbol{I}$. This optimization is solved as follows:

$$\begin{aligned} \min \|\boldsymbol{A} - \boldsymbol{RB}\|_F^2 &= \min \operatorname{Tr}\Big((\boldsymbol{A} - \boldsymbol{RB})^\intercal(\boldsymbol{A} - \boldsymbol{RB})\Big) \\ &= \min \operatorname{Tr}\Big((\boldsymbol{A}^\intercal - \boldsymbol{B}^\intercal\boldsymbol{R}^\intercal)(\boldsymbol{A} - \boldsymbol{RB})\Big) \\ &= \min \operatorname{Tr}\Big(\boldsymbol{A}^\intercal\boldsymbol{A} - 2\boldsymbol{A}^\intercal\boldsymbol{RB} + \boldsymbol{B}^\intercal\boldsymbol{B}\Big) \\ &= \max \operatorname{Tr}\Big(\boldsymbol{A}^\intercal\boldsymbol{RB}\Big) \\ &= \max \operatorname{Tr}\Big(\boldsymbol{RBA}^\intercal\Big) \qquad \text{since trace(PQ) = trace(QP)} \\ \text{using SVD on } \boldsymbol{BA}^\intercal \quad \boldsymbol{BA}^\intercal &= \boldsymbol{UDV}^\intercal \\ &= \max \operatorname{Tr}\Big(\boldsymbol{RUDV}^\intercal\Big) \\ &= \max \operatorname{Tr}\Big(\boldsymbol{V}^\intercal\boldsymbol{RUD}\Big) \\ &= \max \operatorname{Tr}\Big(\boldsymbol{Z(R)D}\Big) \\ &= \max \sum_i z_{ii}d_{ii} \leq \sum_i d_{ii} \quad (\because \boldsymbol{Z(R)}^\intercal\boldsymbol{Z(R)} = \boldsymbol{I}) \end{aligned}$$

The maximum is achieved for $\boldsymbol{Z(R)} = \boldsymbol{I}$, i.e.,

$$\boldsymbol{R} = \boldsymbol{VU}^\intercal$$

This algorithm is used in the **Tomography under unknown angles for 3D images**, where we try to estimate the rotation matrix $\boldsymbol{R}$. This problem is also known as **Orthogonal Procrustes** problem.

5

5.a

The problem of hyperspectral unmixing is defined using the following constrained convex optimization problem:

$$\hat{\boldsymbol{s}}[n] = \arg\min_{\boldsymbol{s}[n]\in\mathcal{S}} \|\boldsymbol{y}[n] - \boldsymbol{A}\boldsymbol{s}[n]\|_2^2\,, \text{ such that } \sum_{i=1}^{N} s_i[n] = 1, s_i[n] \geq 0, i = 1, \ldots, N,\ n = 1, \ldots, L$$

where

- $\boldsymbol{y}[n] = [y_1[n], y_2[n], \ldots, y_M[n]]^\intercal \in \mathbb{R}^M$ is a vector of hyperspectral camera measurements where $y_m[n]$ denotes the camera's measurement at spectral band m and pixel n. There are a total of M spectral bands and L pixels.
- $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N] \in \mathbb{R}^{M\times N}$ is called as the *endmember matrix* and each $\boldsymbol{a_i}$ is called *endmember signature vector* where N is the total number of endmembers or the materials in the scene.
- $\boldsymbol{s}[n] = [s_1[n], \ldots, s_N[n]]^\intercal \in \mathbb{R}^N$ is called the *abundance vector* at pixel n where $s_i[n]$ describes the contribution of material i at pixel n.

5.b

We have the block model

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{S}, \boldsymbol{S} \in \mathbb{R}^{K\times L}, \boldsymbol{Y} \in \mathbb{R}^{M\times L}$$

If S denotes the set of indices of active materials in the measured data $\boldsymbol{Y}$, and if $\boldsymbol{A}_S$ denotes the matrix made from the columns of $\boldsymbol{A}$ indexed by S. If the set of true abundance maps $\{\boldsymbol{s}^i\}_{i\in S}$ are assumed to be linearly independent, then $\mathcal{R}(\boldsymbol{Y}) = \mathcal{R}(\boldsymbol{A}_S)$, where $\mathcal{R}$ denotes the range space of its argument. This gives us equation (40):

$$\|\boldsymbol{S}\|_{\text{row - 0}} < \text{spark}(\boldsymbol{A}) - 1$$

Here spark($\boldsymbol{A}$) denotes the smallest number of linearly dependent columns of $\boldsymbol{A}$ and $\|\boldsymbol{S}\|_{\text{row - 0}}$ denotes the number of non-zero rows in $\boldsymbol{S}$. So, the problem requirements that we have are stated as follows:

- $\boldsymbol{A} \geq 0$, $\boldsymbol{S} \geq 0$ (based on physical constraints) and $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{S}$.
- As $\mathcal{R}(\boldsymbol{Y}) = \mathcal{R}(\boldsymbol{A}_S)$, we are trying to represent the columns of $\boldsymbol{Y}$ in the basis formed using columns of $\boldsymbol{A}$.
- As per the inequality mentioned above, we have $K < M$

All the above requirements are captured by performing NMF on $\boldsymbol{Y}$.

5.c

Standard NMF may not guarantee solution uniqueness. This is a serious issue to the blind HU application, since it means that an NMF solution may not be necessarily be the true endmembers and abundances, even in the noiseless case. Hence, for the case of blind HU, NMF is modified to fit the problem better. The unified NMF-based blind HU problem can be formulated as follows:

$$\min_{\boldsymbol{A}\geq 0, \boldsymbol{S}\in\mathcal{S}^L} \|\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{S}\|_F^2 + \lambda \cdot g(\boldsymbol{A}) + \mu \cdot h(\boldsymbol{S})$$

where $\mathcal{S}^L = \{\boldsymbol{S}|\boldsymbol{s}[n] \geq \boldsymbol{0}, \boldsymbol{1}^\mathsf{T}\boldsymbol{s}[n] = 1, 1 \leq n \leq L\}$, g and h are regularizers and λ and μ are positive constants. The following two variations of blind HU are explained:

1. **Iterated Constrained Endmember**(ICE):
   Here, $h(\boldsymbol{S}) = 0$ and
   $$g(\boldsymbol{A}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2^2$$
   which is the sum of differences between the vertices.

2. **Sparsity Promoting Iterated Constrained Endmember**(SPICE):
   $$g(\boldsymbol{A}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2^2$$
   $$h(\boldsymbol{S}) = \sum_{i=1}^{N} \gamma_i \left\|\boldsymbol{s}^i\right\|_1$$
   Here, we promote sparsity on the columns of $\boldsymbol{S}$