Multilinear Filtering of Nonstationary Array Data and Inversion

J. A. Rod Blais
Dept. of Geomatics Engineering
Pacific Institute for the Mathematical Sciences
University of Calgary, Calgary, AB

blais@ucalgary.ca  www.ucalgary.ca/~blais
Outline

• Introduction
• Linear Filtering of Digital Arrays
• Filtering of Nonstationary Digital Arrays
• Inversion or Deconvolution Problems
• Spherical Filtering Applications
• General Least-Squares Solutions
• Concluding Remarks
Introduction

• Filtering is very common in data processing
• Linear filtering is a mathematical convolution of arrays
• Nonstationary digital arrays often require nonlinear filters
• Multilinear filters can often be used in nonstationary cases
• Seismic inversion problems provided the initial challenge
• Optimization of seismic inversion computations is ongoing
• Spherical applications present different challenges
Linear Filtering as Convolution

One-Dimensional Case:

- By definition: \[ [1\ 2\ 3] \ast [4\ 5] = [4\ 13\ 22\ 15] \]
- Using polynomial products: \({1+2z+3z^2}\cdot{4+5z} = 4+13z+22z^2+15z^3\)
- Using dyadic products: \[
\begin{pmatrix}
4 & 8 & 12 \\
5 & 10 & 15
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
4 & 13 & 22 & 15
\end{pmatrix}
\]
- Using circulant matrices:
\[
\begin{pmatrix}
1 & 0 & 3 & 2 \\
2 & 1 & 0 & 3 \\
3 & 2 & 1 & 0 \\
0 & 3 & 2 & 1
\end{pmatrix}
\begin{pmatrix}
4 \\
5 \\
0 \\
0
\end{pmatrix}
= \begin{pmatrix}
4 \\
13 \\
22 \\
15
\end{pmatrix}
\]
Linear Filtering of Digital Arrays

Example of simple (e.g. mean) filtering of a regular array:

\[
\begin{pmatrix}
... & ... & ... & ... & ... & ... \\
... & a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} & ... & ... \\
... & a_{i,j-1} & a_{i,j} & a_{i,j+1} & ... & ... \\
... & a_{i+1,j-1} & a_{i+1,j} & a_{i+1,j+1} & ... & ... \\
... & ... & ... & ... & ... & ...
\end{pmatrix}
\begin{pmatrix}
m_{1,1} & m_{1,0} & m_{1,-1} \\
m_{0,1} & m_{0,0} & m_{0,-1} \\
m_{-1,1} & m_{-1,0} & m_{-1,-1}
\end{pmatrix}
= \begin{pmatrix}
... & ... & ... & ... & ... & ...
\end{pmatrix}
\begin{pmatrix}
a_{i,j} \\
a_{i,j}
\end{pmatrix}
\]

\[a_{ij} \ast m_{kl} = \hat{a}_{ij}\ i.e.\ \sum_{k=-1}^{1} \sum_{l=-1}^{1} a_{i-k,j-l} m_{kl} = \hat{a}_{ij}\ for\ all\ i\ and\ j\]

which requires a zero-border or operations modulo N with finite N.
Linear Filtering of Stationary Arrays

General linear filtering as convolution

\[ \sum_k \sum_l a_{i-k,j-l} m_{kl} = \tilde{a}_{ij} \quad \text{for all } i \text{ and } j \]

assuming a zero-border or 'modulo' operations in finite applications.

This can be written elementwise simply as:

\[ a_{ij} * m_{kl} = \tilde{a}_{ij} \quad \text{i.e.} \quad a_{i-k,j-l} m_{kl} = \tilde{a}_{ij} \quad \text{for all } i \text{ and } j \]

and more generally,

\[ a_{ijkl} * m_{rs} = \tilde{a}_{ijkl} \quad \text{i.e.} \quad a_{ijk-r,l-s} m_{rs} = \tilde{a}_{ijkl} \quad \text{for all } i, j, k, l \]
Linear Filtering of Nonstationary Arrays

General linear filtering as tensor product

\[ \sum_k \sum_l a_{ijkl} m_{kl} = \hat{a}_{ij} \quad \text{for all } i \text{ and } j \]

which also requires a zero-border or 'modulo' operations in finite applications.

Using the summation over repeated indices, this can be written elementwise simply as:

\[ a_{ijkl} \otimes m_{kl} = \hat{a}_{ij} \quad \text{i.e.} \quad a_{ijkl} m_{kl} = \hat{a}_{ij} \quad \text{for all } i \text{ and } j \]

More generally,

\[ a_{ijklrs} \otimes m_{rs} = \hat{a}_{ijkl} \quad \text{i.e.} \quad a_{ijklrs} m_{rs} = \hat{a}_{ijkl} \quad \text{for all } i, j, k, l \]
Linear Algebraic Formulations

Filtering or Convolutions for Stationary Case:
• matrix formulation:  \( a_{i-k,j-l} m_{kl} = \hat{a}_{ij} \) for all \( i \) and \( j \)
• with equispaced array data: direct and inverse solutions using Fourier transforms (FFTs)

Filtering or Convolutions for Nonstationary Case:
• tensor formulation:  \( a_{ijkl} m_{kl} = \hat{a}_{ij} \) for all \( i \) and \( j \)
• matrix formulation often possible through the so-called 'unwrapping of inner matrices' (see e.g. [Clearbout, 1998])
• for systems such as  \( a_{ijkl} \) corresponding to multi-channel image filtering and the like…
Seismic Inversion Problems

Given a Kirchhoff operator $G$ and an unknown reflectivity model $m$ for array data $d$, the multilinear tensor equation $G \otimes m = d$, where

$G = \left[ g_{ijkl} \mid i = 1, \ldots, I; j = 1, \ldots, J; k = 1, \ldots, K; l = 1, \ldots, L \right]$

$m = \left[ m_{kl} \mid k = 1, \ldots, K; l = 1, \ldots, L \right]$

$d = \left[ d_{ij} \mid i = 1, \ldots, I; j = 1, \ldots, J \right]$

and hence

$$\sum_{k=1}^{K} \sum_{l=1}^{L} g_{ijkl} m_{kl} = d_{ij} \quad \text{for} \quad i = 1, \ldots, I \quad \text{and} \quad j = 1, \ldots, J$$

Or simply, elementwise,

$$g_{ijkl} m_{kl} = d_{ij} \quad \text{assuming that repeated indices are summed over.}$$
Example of Kirchhoff Operator

Figure 2: a) Non-zero elements of matrix $G$, b) Close up of the first 250 rows and columns.

Source: [Yousefzadeh & Bancroft, 2012]
Inversion of Seismic Modeling Equations

\[
\begin{pmatrix}
    g_{1111} & \ldots & g_{111L} \\
    \ldots & \ldots & \ldots \\
    g_{11K1} & \ldots & g_{11KL} \\
\end{pmatrix}
\begin{pmatrix}
    g_{1211} & \ldots & g_{121L} \\
    \ldots & \ldots & \ldots \\
    g_{12K1} & \ldots & g_{12KL} \\
\end{pmatrix}
\begin{pmatrix}
    g_{1J11} & \ldots & g_{1J1L} \\
    \ldots & \ldots & \ldots \\
    g_{1JK1} & \ldots & g_{1JKL} \\
\end{pmatrix}
\begin{pmatrix}
    m_{11} & \ldots & m_{1L} \\
    \ldots & \ldots & \ldots \\
    m_{K1} & \ldots & m_{KL} \\
\end{pmatrix}
=\begin{pmatrix}
    d_{11} & \ldots & d_{1J} \\
    \ldots & \ldots & \ldots \\
    d_{11} & \ldots & d_{1J} \\
\end{pmatrix}
\]
Explicitly, for the ij-th submatrix:

\[
\begin{pmatrix}
g_{ij11} & \cdots & g_{ij1L} \\
\vdots & \ddots & \vdots \\
g_{ijK1} & \cdots & g_{ijKL}
\end{pmatrix}
\otimes
\begin{pmatrix}
m_{11} & \cdots & m_{1L} \\
\vdots & \ddots & \vdots \\
m_{K1} & \cdots & m_{KL}
\end{pmatrix}
= d_{ij}
\]

that is,

\[
g_{ij11} m_{11} + g_{ij12} m_{12} + g_{ij13} m_{13} + \cdots + g_{ijKL} m_{KL} = d_{ij}
\]

implying a linear system \( A x = B \) (assumed over-determined) of \( I \times J \) equations for \( K \times L \) unknowns in the inverse problem.

Note that higher dimensional modeling (e.g. in 3D or 4D) can be analogously formulated with higher rank tensors of the arrays.
Notice that the $ij$-th subarray is not necessarily a matrix:

$$
\begin{pmatrix}
... \\
... \ g_{ijkl} ... \\
... \\
\end{pmatrix}
\otimes
\begin{pmatrix}
... \\
... \ m_{kl} ... \\
... \\
\end{pmatrix} = d_{ij}
$$

that is,

$$
... + g_{ijkl} m_{kl} + ... = d_{ij}
$$

implying a linear system $A x = B$ (assumed over-determined) of equations for $m_{kl}$ unknowns in the inverse problem.

Note that higher dimensional modeling (e.g. in 3D or 4D) can be analogously formulated with higher rank tensors of the arrays.
Spherical Filtering Applications

Case 1: Partial Spherical Applications

Planar situation by appropriate mapping

Computations using FFTs as discussed before

Commutative i.e. $a_{ij} \times b_{kl} = b_{kl} \times a_{ij}$

Case 2: Global Spherical Applications

Gridding as equiangular or near equispaced ([Blais, 2011a & b])

Computations using SHTs (i.e. Spherical Harmonic Transforms)

Non-commutative i.e. $a_{ij} \times b_{kl} \neq b_{kl} \times a_{ij}$ in general!
Spherical Harmonic Transforms

Given an arbitrary array of spherical data \{d_{jk} \mid j=1,J \& k=1,K\}, its spherical harmonic representation is the following

\[
d_{jk} = \sum_{n=0}^{N-1} \sum_{m=0}^{n} (c_{nm} \cos m\lambda_k + s_{nm} \sin m\lambda_k) P_{nm}(\cos \theta_j)
\]

and setting

\[
P_{nmj}^{\cos} \equiv \cos m\lambda_k P_{nm}(\cos \theta_j) \quad \text{and} \quad P_{nmj}^{\sin} \equiv \sin m\lambda_k P_{nm}(\cos \theta_j)
\]

one can write the tensor formulation

\[
\begin{pmatrix}
P_{00j}^{\cos} & P_{10j}^{\sin} & P_{10j}^{\sin} & \cdots \\
P_{10j}^{\cos} & P_{11j}^{\cos} & P_{21j}^{\sin} & \cdots \\
P_{20j}^{\cos} & P_{21j}^{\cos} & P_{22j}^{\cos} & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{pmatrix} \otimes
\begin{pmatrix}
c_{00} & s_{10} & s_{20} & \cdots \\
c_{10} & c_{11} & s_{21} & \cdots \\
c_{20} & c_{21} & c_{22} & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{pmatrix} = d_{jk}
\]
Fast Spherical Harmonic Transforms

For an equilongitudinal array of spherical data \( \{d_{jk} \mid j=1,J \& k=1,2N\} \), considering the FFTs of the rows of data corresponding to the parallels on the sphere, the previous equations reduce to

\[
u_m(\theta_j) + iv_m(\theta_j) = \sum_{n=m}^{N-1} \sum_{j=0}^{2N-1} q_j [u_m(\theta_j) + iv_m(\theta_j)] P_{nm}(\cos \theta_j)
\]

For equiangular array data, following FFTs of the rows, a Chebychev quadrature can be used on the columns to give the inversion results

\[
c_{nm} + is_{nm} = \sum_{j=0}^{2N-1} q_j [u_m(\theta_j) + iv_m(\theta_j)] P_{nm}(\cos \theta_j)
\]

Details with simulations can be found in [Blais, 2011a and b].
General Least-Squares Solutions

For such an overdetermined linear complex system $A x = b$, the simplest normal equations are $A^*A x = A^*b$, or with observational weights $P > 0$, these become $A^*PA x = A^*Pb$, and with parameter weights $Q \geq 0$, $(A^*PA+Q) x = A^*Pb$

Numerically, the Cholesky Square-Root Algorithm is among the best solution method requiring about $O(N^3)$ operations for order $N$.

Notice that a direct solution requires only $1/3$ of the operations using matrix inversion and simplifications are possible with band limited systems i.e. $O(NB^2)$ for bandwidth $B < N/2$. 
Concluding Remarks

• Linear filtering is a simple convolution in stationary cases
• Linear filtering becomes tensorial in nonstationary applications
• The corresponding inversions are different algebraic systems
• Kirchhoff model for seismic imaging is a multilinear system
• Least squares are usually implemented in those inversions
• For ill-conditioned systems, regularization can be exploited
• Spherical filtering is generally more complex and limited
• Formulations and interpretations often need experience!