On the Impact of Prony’s Method

Gerlind Plonka

University of Göttingen

Approximation Theory 16

Nashville, May 2019
Outline

- Original Prony method: Reconstruction of Sparse Exponential Sums
- Maximum Likelihood Modification of Prony’s Method
- Prony’s Method Based on the Shift Operator
- Prony’s Method Based on the Differential Operator
- Generalized Operator Based Prony Method

Collaborations

Thomas Peter, Manfred Tasche, Marius Wischerhoff, Ingeborg Keller, Kilian Stampfer, Ran Zhang
Prony Method: Reconstruction of Sparse Exponential Sums

Function

\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \]

We have \( M, f(\ell), \ell = 0, \ldots, 2M - 1 \)

We want \( c_j, \alpha_j \in \mathbb{C}, \) where \(-\pi \leq \text{Im} \alpha_j < \pi, \) \( j = 1, \ldots, M.\)
Function
\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \]

We have \( M, f(\ell), \ell = 0, \ldots, 2M - 1 \)

We want \( c_j, \alpha_j \in \mathbb{C}, \text{ where } -\pi \leq \text{Im} \alpha_j < \pi, j = 1, \ldots, M. \)

Consider the **Prony polynomial**

\[ P(z) := \prod_{j=1}^{M} (z - e^{\alpha_j}) = \sum_{\ell=0}^{M} p_{\ell} z^{\ell} \]

with unknown parameters \( \alpha_j \) and \( p_M = 1. \)

\[ \sum_{\ell=0}^{M} p_{\ell} f(\ell + m) = \sum_{\ell=0}^{M} p_{\ell} \sum_{j=1}^{M} c_j e^{\alpha_j (\ell+m)} = \sum_{j=1}^{M} c_j e^{\alpha_j m} \sum_{\ell=0}^{M} p_{\ell} e^{\alpha_j \ell} \]

\[ = \sum_{j=1}^{M} c_j e^{\alpha j m} P(e^{\alpha_j}) = 0, \quad m = 0, \ldots, M - 1. \]
Reconstruction Algorithm

**Input:** \( f(\ell), \ell = 0, \ldots, 2M - 1 \)

- Solve the Hankel system

\[
\begin{pmatrix}
  f(0) & f(1) & \ldots & f(M - 1) \\
  f(1) & f(2) & \ldots & f(M) \\
  \vdots & \vdots & \ddots & \vdots \\
  f(M - 1) & f(M) & \ldots & f(2M - 2)
\end{pmatrix}
\begin{pmatrix}
  p_0 \\
  p_1 \\
  \vdots \\
  p_{M-1}
\end{pmatrix}
= -
\begin{pmatrix}
  f(M) \\
  f(M + 1) \\
  \vdots \\
  f(2M - 1)
\end{pmatrix}
\]

- Compute the zeros of the Prony polynomial \( P(z) = \sum_{\ell=0}^{M} p_{\ell} z^{\ell} \) and extract the parameters \( \alpha_j \) from its zeros \( z_j = e^{\alpha_j}, j = 1, \ldots, M \).

- Compute \( c_j \) solving the linear system

\[
f(\ell) = \sum_{j=1}^{M} c_{j} e^{\alpha_j \ell}, \quad \ell = 0, \ldots, 2M - 1.
\]

**Output:** Parameters \( \alpha_j \) and \( c_j, j = 1, \ldots, M \).
(Almost) Equivalent Models

If we can reconstruct

\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x}, \]

then we can also reconstruct

\[ g(t) = \sum_{j=1}^{M} c_j \delta(t - t_j) \Rightarrow \hat{g}(x) = \sum_{j=1}^{M} c_j e^{-i\alpha_j x} \]

\[ g(t) = \sum_{j=1}^{M} c_j \phi(t - t_j) \Rightarrow \hat{g}(x) = \left( \sum_{j=1}^{M} c_j e^{-i\alpha_j x} \right) \hat{\phi}(x) \]

\[ g(t) = \sum_{j=1}^{M} \frac{c_j}{t - \alpha_j} \Rightarrow \mathcal{L}^{-1}(g)(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \]
[Prony] (1795): Reconstruction of difference equations
[Schmidt] (1979): MUSIC (Multiple Signal Classification)
[Hua, Sakar] (1990): Matrix-pencil method
[Stoica, Moses] (2000): Annihilating filters
[Vetterli, Marziliano, Blu] (2002): Finite rate of innovation signals
[Peter, Plonka] (2013): Generalized Prony Method

Sidi ('75,'82,'85); Golub, Milanfar, Varah ('99); Maravić, Vetterli ('04);
Elad, Milanfar, Golub ('04); Beylkin, Monzon ('05,'10);
Andersson, Carlsson, de Hoop ('10), Berent, Dragotti, Blu ('10),
Batenkov, Sarg, Yomdin ('12,'13); Filbir, Mhaskar, Prestin ('12);
Peter, Potts, Tasche ('11,'12,'13); Plonka, Wischerhoff ('13);
Plonka, Tasche ('14); Kunis, Peter, Römer, von der Ohe ('16);
Wei, Dragotti ('16); Sauer ('17); Cuyt, Lee ('17), Mourrain ('17), . . .
Very incomplete list !!!
Other Talks in this Conference

- A. Aldroubi, L. Huang, K. Kornelson, and I. Krishtal: A Prony-Laplace Method for Identifying Burst-like Forcing Terms
- G. Plonka, K. Stampfer and I. Keller: Reconstruction of Non-Stationary Signals by the Generalized Prony Method
- Sui Tang: Recovery of Linear Dynamics from Undersampled Time Series Data
- Z.M. Wu and R. Zhang: Learning Physics by Data for the Motion of a Sphere Falling in a Non-Newtonian Fluid
Maximum Likelihood Modification of Prony’s Method

Let \( y = (y_k)_{k=0}^L \in \mathbb{C}^{L+1} \) be a given.

**Goal:** Approximate \( y \) by \( f = (f_k)_{k=0}^L \in \mathbb{C}^{L+1} \) where

\[
f_k = \sum_{j=1}^M d_j e^{\alpha_j k} = \sum_{j=1}^M d_j z^k_j, \quad k = 0, \ldots, L, \quad M \leq L/2
\]

with \( d_j, z_j = e^{\alpha_j} \in \mathbb{C}, j = 1, \ldots, M \), such that

\[
\|y - f\|_2^2 = \sum_{k=0}^L |y_k - f_k|^2
\]

is minimal.

**Survey:** Zhang & Plonka ’19
Maximum Likelihood Modification of Prony’s Method

With \( \mathbf{d} := (d_1, \ldots, d_M)^T \), \( \mathbf{z} := (z_1, \ldots, z_M)^T \), and

\[
\mathbf{V}_z := \begin{pmatrix}
1 & 1 & \ldots & 1 \\
z_1 & z_2 & \ldots & z_M \\
z_1^2 & z_2^2 & \ldots & z_M^2 \\
\vdots & \vdots & \ddots & \vdots \\
z_1^L & z_2^L & \ldots & z_M^L 
\end{pmatrix} \in \mathbb{C}^{(L+1) \times M},
\]

we have

\[
f = \mathbf{V}_z \mathbf{d}.
\]

Thus, we want to solve the nonlinear least squares problem

\[
\arg\min_{\mathbf{z}, \mathbf{d} \in \mathbb{C}^M} \| \mathbf{y} - \mathbf{V}_z \mathbf{d} \|^2 = \arg\min_{\mathbf{z}, \mathbf{d} \in \mathbb{C}^M} \sum_{k=0}^{L} |y_k - \sum_{j=1}^{M} d_j z_j^k|^2.
\]
Variable projection gives

\[ d = V_z^+ y = [V_z^* V_z]^{-1} V_z^* y. \]

Thus, solve

\[
\argmin_{z \in \mathbb{C}^M} \| y - V_z V_z^+ y \|_2^2 = \argmax_{z \in \mathbb{C}^M} (y^* V_z V_z^+ y).
\]
Maximum Likelihood Modification of Prony’s Method

Variable projection gives

\[ d = V_z^+ y = [V_z^* V_z]^{-1} V_z^* y. \]

Thus, solve

\[
\arg\min_{z \in \mathbb{C}^M} \| y - V_z V_z^+ y \|_2^2 = \arg\max_{z \in \mathbb{C}^M} (y^* V_z V_z^+ y).
\]

Let

\[
X_p^T := \begin{pmatrix}
p_0 & p_1 & \ldots & p_M \\
p_0 & p_1 & \ldots & p_M \\
\vdots & \ddots & \ddots & \vdots \\
p_0 & p_1 & \ldots & p_M
\end{pmatrix}
\]

such that \( X_p^T V_z = 0 \). Then, we have to solve

\[
\tilde{p} := \arg\min_{p \in \mathbb{C}^{M+1}} p \in \mathbb{C}^{M+1} \| p \|_2 = 1 \ y^* \ X_p \ X_p^+ y.
\]
Theorem

For given \( y = (y_0, \ldots, y_L)^T \) the vectors \( \tilde{z} \) and \( \tilde{d} \) solving

\[
\min_{z,d} \| y - Vz d \|_2^2
\]

are obtained by:

1. Solve \( \tilde{p} = \arg\min_{p \in \mathbb{C}^{M+1}, \|p\|_2 = 1} y^* \bar{X}_p \bar{X}_p^+ y = \arg\min_{p \in \mathbb{C}^{M+1}, \|p\|_2 = 1} p^* H_y [X_p^T X_p]^{-1} H_y p. \)

2. Compute the vector of zeros \( \tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_M)^T \) of \( p(z) = \sum_{k=0}^{M} \tilde{p}_k z^k \)

with \( \tilde{p} = (\tilde{p}_0, \ldots, \tilde{p}_M)^T. \)

3. Compute

\[
\tilde{d} = V_{\tilde{z}}^+ y = [V_{\tilde{z}}^* V_{\tilde{z}}]^{-1} V_{\tilde{z}}^* y.
\]
Maximum Likelihood Modification of Prony’s Method

Solve \( \hat{p} = \arg\min_{p \in \mathbb{C}^{M+1}} \|p\|_2 = 1 p^* H_y [X_p^T \overline{X_p}]^{-1} H_y p. \)

Approaches:

- Pisarenko ’73 method solves only \( \arg\min_{p \in \mathbb{C}^{M+1}} \|p\|_2 = 1 p^* H_y H_y p. \)

- Levenberg-Marquardt Iteration (weighted structured low-rank approximation) (Markovsky & Usevich ’14)

- Iterative Quadratic Maximum Likelihood (IQML): (Bressler & Macovski ’86, Z. Dogan et al. ’15)

\[ p_{j+1} = \arg\min_{p \in \mathbb{C}^{M+1}} \|p\| = 1 p^* H_y [X_{p_j}^T \overline{X_{p_j}}]^{-1} H_y p. \]

- Gradient Condition Reweighting Algorithm (GRA): (Osborne & Smith ’91,’95)

- Simultaneous Minimization (SIMI) (Zhang & Plonka ’19)
Example: Consider

\[ y_k = \exp(0.95kh) + \exp(0.5kh) + \exp(0.2kh) + \epsilon_k \quad k = 0, 1, \ldots, L \]

with \( \epsilon_k \sim N(0, \sigma^2) \), \( \sigma = 0.01 \), \( L = 69 \), \( h = 5/L \).

Results:

<table>
<thead>
<tr>
<th></th>
<th>APM</th>
<th>GRA</th>
<th>IQML</th>
<th>VARPRO</th>
<th>SIMI</th>
<th>IGRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{\alpha}_1 )</td>
<td>12.13</td>
<td>0.957</td>
<td>0.939</td>
<td>0.939</td>
<td>0.953</td>
<td>0.957</td>
</tr>
<tr>
<td>( \tilde{\alpha}_2 )</td>
<td>1.014</td>
<td>0.566</td>
<td>0.4 + 43.4i</td>
<td>0.8 + 43.4i</td>
<td>0.500</td>
<td>0.565</td>
</tr>
<tr>
<td>( \tilde{\alpha}_3 )</td>
<td>0.532</td>
<td>0.214</td>
<td>0.338</td>
<td>0.338</td>
<td>0.165</td>
<td>0.214</td>
</tr>
<tr>
<td>2-error</td>
<td>4.646</td>
<td>0.0013</td>
<td>0.0014</td>
<td>0.0014</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
</tbody>
</table>

Using the correct parameters 0.95, 0.5, 0.2 we get a 2-error 0.0013. The parameter reconstruction is ill-posed but we get very good approximations.
Towards a Generalization of Prony’s Method
Revisiting Prony’s Method Using the Shift Operator

Let $S_h f := f(\cdot + h)$, $h \in \mathbb{R} \setminus \{0\}$. Then

$$(S_h e^{\alpha \cdot})(x) = e^{\alpha (h + x)} = e^{\alpha h} e^{\alpha x} \quad \text{(eigenfunction)}.$$
Revisiting Prony’s Method Using the Shift Operator

Let $S_h f := f(\cdot + h), \quad h \in \mathbb{R} \setminus \{0\}$. Then

$$(S_h e^{\alpha \cdot})(x) = e^{\alpha(h+x)} = e^{\alpha h} e^{\alpha x} \quad \text{(eigenfunction)}.$$ 

For

$$f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \quad \text{with} \quad P(z) := \prod_{j=1}^{M} (z - e^{\alpha_j h}) = \sum_{\ell=0}^{M} p_\ell z^\ell$$

we have

$$P(S_h) f = \sum_{\ell=0}^{M} p_\ell (S_h^\ell f) = \sum_{\ell=0}^{M} p_\ell S_h^\ell \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} = \sum_{\ell=0}^{M} p_\ell \sum_{j=1}^{M} c_j S_h^\ell e^{\alpha_j \cdot}.$$
Let $S_h f := f(\cdot + h), \quad h \in \mathbb{R} \setminus \{0\}$. Then

$$(S_h e^{\alpha \cdot})(x) = e^{\alpha(h+x)} = e^{\alpha h} e^{\alpha x} \quad \text{(eigenfunction)}.$$ 

For

$$f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \quad \text{with} \quad P(z) := \prod_{j=1}^{M} (z - e^{\alpha_j h}) = \sum_{\ell=0}^{M} p_{\ell} z^\ell$$

we have

$$P(S_h f) = \sum_{\ell=0}^{M} p_{\ell} (S_h^\ell f) = \sum_{\ell=0}^{M} p_{\ell} S_h^\ell \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} = \sum_{\ell=0}^{M} p_{\ell} \sum_{j=1}^{M} c_j S_h^\ell e^{\alpha_j \cdot}$$

$$= \sum_{\ell=0}^{M} p_{\ell} \sum_{j=1}^{M} c_j e^{\alpha_j h \ell} e^{\alpha_j \cdot} = \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} \sum_{\ell=0}^{M} p_{\ell} e^{\alpha_j h \ell} = 0.$$
Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the difference equation \( P(S_h) f = 0 \).

Moreover, \( S_h^k P(S_h) f = P(S_h) S_h^k f = \sum_{\ell=0}^{M} p_{\ell} S_{h}^{\ell + k} f = 0, \quad k \in \mathbb{Z} \).
Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the difference equation \( P(S_h)f = 0 \).

Moreover,

\[
S_h^k P(S_h)f = P(S_h)S_h^k f = \sum_{\ell=0}^{M} p_\ell S_h^{\ell+k} f = 0, \quad k \in \mathbb{Z}.
\]

With the point evaluation functional \( F_0 f := f(0) \),

\[
F_0(S_h^k P(S_h)f) = \sum_{\ell=0}^{M} p_\ell F_0(S_h^{\ell+k} f) = \sum_{\ell=0}^{M} p_\ell f(h(\ell + k)) = 0, \quad k \in \mathbb{Z}.
\]

We can compute the coefficients \( p_\ell \) of the Prony polynomial from \( M \) of these equations, i.e., using \( f(hk), k = 0, \ldots, 2M - 1 \).
Change the Sampling Scheme

Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the difference equation \( P(S_h)f = 0 \).

Moreover, for each linear operator \( T : C^\infty(\mathbb{R}) \mapsto C^\infty(\mathbb{R}) \)

\[
T^k P(S_h)f = T^k P(S_h)f = \sum_{\ell=0}^{M} p_\ell T^k S_h^\ell f = 0, \quad k \in \mathbb{Z}.
\]
Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the difference equation \( P(S_h)f = 0 \).

Moreover, for each linear operator \( T : C^\infty(\mathbb{R}) \hookrightarrow C^\infty(\mathbb{R}) \):

\[
T^k P(S_h)f = T^k P(S_h)f = \sum_{\ell=0}^{M} p_\ell T^k S_h^\ell f = 0, \quad k \in \mathbb{Z}.
\]

With the linear functional \( F : C^\infty(\mathbb{R}) \hookrightarrow \mathbb{C} \):

\[
F(T^k P(S_h)f) = \sum_{\ell=0}^{M} p_\ell F(T^k S_h^\ell f) = \sum_{\ell=0}^{M} p_\ell F(T^k f(h(\ell + \cdot))) = 0, \quad k \in \mathbb{Z}.
\]

We can compute the coefficients \( p_\ell \) of the Prony polynomial from \( M \) of these equations, i.e., using \( F(T^k S_h^\ell f), \quad \ell = 0, \ldots, M, \quad k = 1, \ldots, M \).
Revisiting Prony’s Method Using the Shift Operator

**Example:** Choose $h = 1, \ T = S_{h/2} = S_{1/2}, \ F = F_0$, then the linear system reads (for even $M$)

\[
\begin{pmatrix}
    f(0) & f(1) & \ldots & f(M - 1) \\
    f(\frac{1}{2}) & f(\frac{3}{2}) & \ldots & f(\frac{M-1}{2}) \\
    f(1) & f(2) & \ldots & f(M) \\
    \vdots & \vdots & & \vdots \\
    f(\frac{M}{2} - 1) & f(\frac{M}{2}) & \ldots & f(\frac{3M}{2} - 2)
\end{pmatrix}
\begin{pmatrix}
p_0 \\
p_1 \\
p_2 \\
p_{M-1}
\end{pmatrix} = -
\begin{pmatrix}
f(M) \\
f(M + \frac{1}{2}) \\
f(M + 1) \\
\vdots \\
f(\frac{3M}{2} - 1)
\end{pmatrix}
\]

This matrix is not longer of Hankel form.
Let $\frac{d}{dx} : C^\infty(\mathbb{R}) \mapsto C^\infty(\mathbb{R})$, $\frac{d}{dx} f := f'$. Then

$$(\frac{d}{dx} e^{\alpha \cdot})(x) = \alpha e^{\alpha x} \quad \text{(eigenfunction)}.$$
Let \( \frac{d}{dx} : C^\infty(\mathbb{R}) \mapsto C^\infty(\mathbb{R}) \), \( \frac{d}{dx} f := f' \). Then
\[
\left( \frac{d}{dx} e^{\alpha \cdot} \right)(x) = \alpha e^{\alpha x} \quad \text{(eigenfunction)}.
\]

For
\[
f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \quad \text{with} \quad \tilde{P}(z) := \prod_{j=1}^{M} (z - \alpha_j) = \sum_{\ell=0}^{M} \tilde{p}_\ell z^\ell
\]
we have
\[
\tilde{P} \left( \frac{d}{dx} \right) f = \sum_{\ell=0}^{M} \tilde{p}_\ell \left( \frac{d}{dx} \right)^\ell f = \sum_{\ell=0}^{M} \tilde{p}_\ell \left( \frac{d}{dx} \right)^\ell \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} = \sum_{\ell=0}^{M} \tilde{p}_\ell \sum_{j=1}^{M} c_j \left( \frac{d}{dx} \right)^\ell e^{\alpha_j \cdot}.
\]
Revisiting Prony’s Method Using the Differential Operator

Let \( \frac{d}{dx} : C^\infty(\mathbb{R}) \leftrightarrow C^\infty(\mathbb{R}), \quad \frac{d}{dx} f := f' \). Then

\[
\left( \frac{d}{dx} e^{\alpha \cdot} \right)(x) = \alpha e^{\alpha x} \quad \text{(eigenfunction)}.
\]

For

\[
f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \quad \text{with} \quad \tilde{P}(z) := \prod_{j=1}^{M} (z - \alpha_j) = \sum_{\ell=0}^{M} \tilde{p}_\ell z^\ell
\]

we have

\[
\tilde{P} \left( \frac{d}{dx} \right) f = \sum_{\ell=0}^{M} \tilde{p}_\ell \left( \frac{d}{dx} \right)^\ell f = \sum_{\ell=0}^{M} \tilde{p}_\ell \left( \frac{d}{dx} \right)^\ell \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} = \sum_{\ell=0}^{M} \tilde{p}_\ell \sum_{j=1}^{M} c_j \left( \frac{d}{dx} \right)^\ell e^{\alpha_j \cdot} = \sum_{\ell=0}^{M} \tilde{p}_\ell \sum_{j=1}^{M} c_j \alpha_j^\ell e^{\alpha_j \cdot} = \sum_{j=1}^{M} c_j e^{\alpha_j \cdot} \sum_{\ell=0}^{M} \tilde{p}_\ell \alpha_j^\ell = 0.
\]
Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the differential equation \( \tilde{P}(\frac{d}{dx})f = 0 \).

Moreover,

\[
(\frac{d}{dx})^k \tilde{P}(\frac{d}{dx})f = \sum_{\ell=0}^{M} \tilde{p}_\ell (\frac{d}{dx})^{\ell+k} f = 0, \quad k \in \mathbb{Z}.
\]
Thus, \( f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j x} \) solves the differential equation \( \tilde{P}(\frac{d}{dx})f = 0 \).

Moreover,

\[
(\frac{d}{dx})^k \tilde{P}(\frac{d}{dx})f = \sum_{\ell=0}^{M} \tilde{p}_\ell (\frac{d}{dx})^{\ell+k} f = 0, \quad k \in \mathbb{Z}.
\]

With the point evaluation functional \( F_0 f := f(0) \),

\[
F_0((\frac{d}{dx})^k \tilde{P}(\frac{d}{dx})f) = \sum_{\ell=0}^{M} \tilde{p}_\ell F_0((\frac{d}{dx})^{\ell+k} f) = \sum_{\ell=0}^{M} \tilde{p}_\ell f^{(\ell+k)}(0) = 0, \quad k \in \mathbb{Z}.
\]

We can compute the coefficients \( \tilde{p}_\ell \) of the Prony polynomial from \( M \) of these equations, i.e., using \( f^{(k)}(0), \ k = 0, \ldots, 2M - 1 \).
Switch Between Operators with the Same Eigenfunctions

What is the connection between Prony’s method using the shift operator or the differential operator?

We have

$$\frac{d}{dx} e^{\alpha \cdot} = \alpha e^{\alpha \cdot}, \quad S_h e^{\alpha \cdot} = e^{\alpha h} e^{\alpha \cdot}.$$ 

Obviously, the spectra are connected by the map $\exp(h \cdot) : \alpha \mapsto e^{h \alpha}$. 
What is the connection between Prony’s method using the shift operator or the differential operator?

We have

$$\frac{d}{dx} e^{\alpha x} = \alpha e^{\alpha x}, \quad S_h e^{\alpha x} = e^{\alpha h} e^{\alpha x}.$$  

Obviously, the spectra are connected by the map \( \exp(h \cdot) : \alpha \mapsto e^{h\alpha} \).

Moreover, for all monomials \( x^m \),

$$\exp(h \frac{d}{dx}) x^m = \sum_{k=0}^{\infty} \frac{h^k}{k!} (\frac{d}{dx})^k x^m = \sum_{k=0}^{m} \frac{h^k}{k!} \frac{m!}{(m-k)!} x^{m-k}$$  

$$= \sum_{k=0}^{m} \binom{m}{k} h^k x^{m-k} = (x + h)^m = S_h x^m.$$  

Thus,

$$\exp(h \frac{d}{dx}) f(x) = S_h f(x).$$
Generalized Prony method (Peter & Plonka ’13)

\( \mathcal{V} \): normed vector space
\( \mathcal{A} : \mathcal{V} \to \mathcal{V} \) linear operator
\( \{ \nu_\lambda : \lambda \in \sigma(\mathcal{A}) \} \) set of eigenfunctions of \( \mathcal{A} \) to pairwise different eigenvalues \( \lambda \in \sigma(\mathcal{A}) \subset \mathbb{C} \),

\[ \mathcal{A} \nu_\lambda = \lambda \nu_\lambda. \]

Let

\[ f = \sum_{\lambda \in \Lambda_f} c_\lambda \nu_\lambda, \quad \Lambda_f \subset \sigma(\mathcal{A}) \quad \text{with} \quad |\Lambda_f| = M, \ c_\lambda \in \mathbb{C}. \]

Let \( G : \mathcal{V} \to \mathbb{C} \) be a linear functional with \( G(\nu_\lambda) \neq 0 \) for all \( \lambda \in \sigma(\mathcal{A}) \).

We have \( M, \ G(\mathcal{A}^\ell f) \) for \( \ell = 0, \ldots, 2M - 1 \)

We want \( \Lambda_f \subset \sigma(\mathcal{A}) \), \( c_\lambda \in \mathbb{C} \) for \( \lambda \in \Lambda_f \)
Generalized Prony Method

Theorem (Peter & Plonka ’13)

The expansion

\[ f = \sum_{\lambda \in \Lambda_f} c_{\lambda} v_{\lambda}, \quad \Lambda_f \subset \sigma(A) \quad \text{with} \quad |\Lambda_f| = M, \ c_{\lambda} \in \mathbb{C}. \]

of eigenfunctions \( v_{\lambda} \) of the linear operator \( A \) can be uniquely recovered by \( G(A^\ell f) \), \( \ell = 0, \ldots, 2M - 1 \), where \( G : V \to \mathbb{C} \) is a linear functional with \( G(v_{\lambda}) \neq 0 \) for all \( \lambda \in \sigma(A) \).
Theorem (Peter & Plonka '13)

The expansion

\[ f = \sum_{\lambda \in \Lambda_f} c_\lambda \, v_\lambda, \quad \Lambda_f \subset \sigma(A) \quad \text{with} \quad |\Lambda_f| = M, \, c_\lambda \in \mathbb{C}. \]

of eigenfunctions \( v_\lambda \) of the linear operator \( A \) can be uniquely recovered by \( G(A^\ell f) \), \( \ell = 0, \ldots, 2M - 1 \), where \( G : V \rightarrow \mathbb{C} \) is a linear functional with \( G(v_\lambda) \neq 0 \) for all \( \lambda \in \sigma(A) \).

Example: Let \( V = C^\infty \), \( A := S_h \) with \( S_h f = f(h + \cdot) \), and

\[ f = \sum_{\lambda \in \Lambda_f} c_\lambda \, v_\lambda = \sum_{\lambda \in \Lambda_f} c_\lambda \, e^{\lambda \cdot}, \]

where \( \Lambda_f \subset \mathbb{R} + i[-\pi, \pi) \). Choose the functional \( Gf := f(0) \), then \( f \) can be recovered from the samples

\[ G(A^\ell f) = S^\ell_h f(0) = f(h\ell), \quad \ell = 0, \ldots, 2M - 1. \]
Assume, you want to recover a sparse expansion

\[ f = \sum_{j=1}^{M} c_j v_j, \quad v_j \in V, \ c_j \in \mathbb{C}. \]

**Idea:**

- Find a linear operator \( A \) such that \( v_j \) are eigenfunctions of \( A \) to pairwise different eigenvalues.
- Check, whether \( G(A^\ell f), \ \ell = 0, \ldots, 2M - 1 \) can be computed from the given information.
- If not, transfer to a different operator \( B = \varphi(A) \) and suitable functionals \( G_k \) such that \( G_k(B^\ell f), \ \ell = 0, \ldots, M, \ k = 1, \ldots, M \) that can be computed from the given information.
- Apply the generalized Prony method to recover \( v_j \) and \( c_j \), \( j = 1, \ldots, M \).
Example:

\[ f(x) = \sum_{j=1}^{M} c_j x^{\alpha_j}, \quad \alpha_j \in \mathbb{R}, \ c_j \in \mathbb{R}. \]

- Find a linear operator \( A \) on \( C^\infty(\mathbb{R}) \):

\[ A f(x) := x f'(x) \quad \Rightarrow \quad A x^{\alpha_j} = \alpha_j x^{\alpha_j}. \]

However, \( A^\ell f \) involves higher order derivatives.
Generalized Operator Based Prony Method

Example:

\[ f(x) = \sum_{j=1}^{M} c_j x^{\alpha_j}, \quad \alpha_j \in \mathbb{R}, \ c_j \in \mathbb{R}. \]

- Find a linear operator \( \mathcal{A} \) on \( C^\infty(\mathbb{R}) \):

\[ \mathcal{A} f(x) := x f'(x) \quad \Rightarrow \quad \mathcal{A} x^{\alpha_j} = \alpha_j x^{\alpha_j}. \]

However, \( \mathcal{A}^\ell f \) involves higher order derivatives.

- Choose \( \varphi(z) := \exp(\tau z) \) with \( \tau \in \mathbb{R} \setminus \{0\} \).

\[
\exp(\tau x \frac{d}{dx}) x^m = \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} (x \frac{d}{dx})^\ell x^m = \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} m^\ell x^m = e^{\tau m} x^m = (e^\tau x)^m.
\]

Thus \( \mathcal{B} f(x) = \varphi(\mathcal{A}) f(x) = f(e^\tau x) \quad \text{(dilation operator)}. \)
Generalized Operator Based Prony Method

Example:

\[ f(x) = \sum_{j=1}^{M} c_j x^{\alpha_j}, \quad \alpha_j \in \mathbb{R}, \ c_j \in \mathbb{R}. \]

- Find a linear operator \( \mathcal{A} \) on \( C^\infty(\mathbb{R}) \):

\[ \mathcal{A} f(x) := x f'(x) \quad \Rightarrow \quad \mathcal{A} x^{\alpha_j} = \alpha_j x^{\alpha_j}. \]

However, \( \mathcal{A}^\ell f \) involves higher order derivatives.

- Choose \( \varphi(z) := \exp(\tau z) \) with \( \tau \in \mathbb{R} \setminus \{0\} \).

\[ \exp(\tau x \frac{d}{dx}) x^m = \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} (x \frac{d}{dx})^\ell x^m = \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} m^\ell x^m = e^{\tau m} x^m = (e^\tau x)^m. \]

Thus \( B f(x) = \varphi(\mathcal{A}) f(x) = f(e^{\tau} x) \) (dilation operator).

- Choose \( f(e^{\ell \tau} x_0), \ \ell = 0, \ldots, 2M - 1 \) to recover \( f \).
Example: dilation operator

Consider

\[ f(x) = \frac{6}{x^9} + \frac{\sqrt{x}}{5} + 1.3x. \]

Choose \( G(f) := f(x_0) \) with \( x_0 = -0.7 - 0.7i \) and \( h = 1.1e^{i/5} \)
Reconstruct \[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j G(x)}, \]
i.e., find \( c_j, \alpha_j, j = 1, \ldots, M \), where \( G \) is differentiable and strictly monotone on \([a, b]\).
Reconstruct

\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j G(x)}, \]

i.e., find \( c_j, \alpha_j, j = 1, \ldots, M \), where \( G \) is differentiable and strictly monotone on \([a, b]\).

Find a linear operator with eigenfunctions \( e^{\alpha_j G(x)} \):

Let \( g(x) := 1/G'(x) \) and

\[ A f(x) := g(x) \frac{d}{dx} f(x). \]

Then

\[ A e^{\alpha G(x)} = g(x) \frac{d}{dx} e^{\alpha G(x)} = \alpha e^{\alpha G(x)}. \]
Differential Operators of First Order and Generalized Shifts

Reconstruct

\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j G(x)}, \]

i.e., find \( c_j, \alpha_j, j = 1, \ldots, M \), where \( G \) is differentiable and strictly monotone on \([a, b]\).

Find a linear operator with eigenfunctions \( e^{\alpha_j G(x)} \):

Let \( g(x) := 1/G'(x) \) and

\[ \mathcal{A}f(x) := g(x) \frac{d}{dx} f(x). \]

Then

\[ \mathcal{A} e^{\alpha G(x)} = g(x) \frac{d}{dx} e^{\alpha G(x)} = \alpha e^{\alpha G(x)}. \]

Using the generalized Prony method, \( f \) can be recovered using

\[ F((g(\cdot) \frac{d}{dx})^k f), \quad k = 0, \ldots, 2M - 1. \]

However, these may be difficult to provide.
Differential Operators of First Order and Generalized Shifts

Change the operator:

\[
\exp(\tau A)f(x) = \exp(\tau g(x) \frac{d}{dx}) f(x)
\]

\[
= \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} \left( g(x) \frac{d}{dx} \right)^\ell \left( \sum_{j=1}^{M} c_j e^{\alpha_j G(x)} \right)
\]

\[
= \sum_{j=1}^{M} c_j \left( \sum_{\ell=0}^{\infty} \frac{\tau^\ell}{\ell!} \alpha_j^\ell \right) e^{\alpha_j G(x)}
\]

\[
= \sum_{j=1}^{M} c_j e^{\alpha_j \tau} e^{\alpha_j G(x)}
\]

\[
= \sum_{j=1}^{M} c_j e^{\alpha_j G(G^{-1}(\tau + G(x)))} = f(G^{-1}(\tau + G(x))).
\]
Theorem (Stampfer & Plonka ’19)

Let

\[ f(x) = \sum_{j=1}^{M} c_j e^{\alpha_j G(x)}, \]

where \( G(x) \) is continuous and monotone on an interval \([a, b]\). Let \( \tau k + G(x_0) \in G([a, b]) \) for \( k = 0, \ldots, 2M - 1 \). Then \( f(x) \) can be uniquely reconstructed from the function samples

\[ f(G^{-1}(\tau k + G(x_0))), \quad k = 0, \ldots, 2M - 1. \]
### Differential Operators of First Order and Generalized Shifts

<table>
<thead>
<tr>
<th>$g(x)$</th>
<th>$G(x)$</th>
<th>eigenfunctions</th>
<th>sampling values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{x}$</td>
<td>$-\frac{1}{2}x^2$</td>
<td>$\exp\left(-\frac{\alpha}{2}x^2\right)$</td>
<td>$f\left(\sqrt{x_0} - k\tau\right)$</td>
</tr>
<tr>
<td>$1$</td>
<td>$x$</td>
<td>$\exp(\alpha x)$</td>
<td>$f\left(\tau k + x_0\right)$</td>
</tr>
<tr>
<td>$x$</td>
<td>$\log(x)$</td>
<td>$x^\alpha$</td>
<td>$f\left(e^{\tau k}x_0\right)$</td>
</tr>
<tr>
<td>$-\sqrt{1-x^2}$</td>
<td>$\arccos x$</td>
<td>$\exp(\alpha \arccos x)$</td>
<td>$f\left(\cos(k\tau + \arccos(x_0))\right)$</td>
</tr>
<tr>
<td>$\sqrt{1-x^2}$</td>
<td>$\arcsin x$</td>
<td>$\exp(\alpha \arcsin x)$</td>
<td>$f\left(\sin(k\tau + \arcsin(x_0))\right)$</td>
</tr>
<tr>
<td>$\frac{1}{\cos(x)}$</td>
<td>$\sin x$</td>
<td>$\exp(\alpha \sin x)$</td>
<td>$f\left(\arcsin(k\tau + \sin(x_0))\right)$</td>
</tr>
<tr>
<td>$-\frac{1}{\sin(x)}$</td>
<td>$\cos x$</td>
<td>$\exp(\alpha \cos x)$</td>
<td>$f\left(\arccos(k\tau + \cos(x_0))\right)$</td>
</tr>
</tbody>
</table>

Examples of operators $A = g(\cdot) \frac{d}{dx}$, corresponding eigenfunctions $\exp(\alpha G(\cdot))$ and sampling values for $k = 0, \ldots, 2M-1$ with sampling parameter $\tau$ to recover expansions $f$. 
Example: Recovery of shifts of Gaussians (Plonka, Stampfer, Keller ’19)

\[ f(x) = \sum_{j=1}^{5} c_j e^{i(x-\alpha_j)^2} \]

<table>
<thead>
<tr>
<th></th>
<th>( j = 1 )</th>
<th>( j = 2 )</th>
<th>( j = 3 )</th>
<th>( j = 4 )</th>
<th>( j = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re ( c_j )</td>
<td>-2.37854</td>
<td>-4.55545</td>
<td>2.54933</td>
<td>-2.57214</td>
<td>-0.57597</td>
</tr>
<tr>
<td>Im ( c_j )</td>
<td>0.75118</td>
<td>-0.56308</td>
<td>0.94536</td>
<td>0.42117</td>
<td>0.73366</td>
</tr>
<tr>
<td>( \alpha_j )</td>
<td>0.64103</td>
<td>-0.18125</td>
<td>-1.50929</td>
<td>-0.53137</td>
<td>-0.23778</td>
</tr>
</tbody>
</table>

Real part
Example: Recovery of shifts of Gaussians

\[ f(x) = \sum_{j=1}^{5} c_j e^{i(x-\alpha_j)^2} \]

<table>
<thead>
<tr>
<th></th>
<th>( j = 1 )</th>
<th>( j = 2 )</th>
<th>( j = 3 )</th>
<th>( j = 4 )</th>
<th>( j = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Re } c_j )</td>
<td>-2.37854</td>
<td>-4.55545</td>
<td>2.54933</td>
<td>-2.57214</td>
<td>-0.57597</td>
</tr>
<tr>
<td>( \text{Im } c_j )</td>
<td>0.75118</td>
<td>-0.56308</td>
<td>0.94536</td>
<td>0.42117</td>
<td>0.73366</td>
</tr>
<tr>
<td>( \alpha_j )</td>
<td>0.64103</td>
<td>-0.18125</td>
<td>-1.50929</td>
<td>-0.53137</td>
<td>-0.23778</td>
</tr>
</tbody>
</table>

imaginary part
Summary

- Prony’s method can be used in many contexts, since sparse representations can be often transformed to the form of exponential sums, e.g.,

\[ g(t) = \sum_{j=1}^{M} c_j \phi(t - t_j) \quad \Rightarrow \quad \hat{g}(x) = \left( \sum_{j=1}^{M} c_j e^{-it_jx} \right) \hat{\phi}(x) \]

- The underlying recovery problem is ill-posed.
- For noisy samples, one should use the modified Prony method.
- Prony’s method can be generalized to recover sparse expansions of eigenfunctions of linear operators.
- One can use different operators with the same (sub)set of eigenfunctions.
- One can employ the generalized Prony method to find new more general sampling schemes.


