University of Wisconsin-Madison | Lecturer: Benjamin Recht

**CS838 Topics In Optimization:**
**Convex Geometry in High-Dimensional Data Analysis**

February 4, 2010 | Scribe: Matt Malloy, Zhiting Xu

---

# Lecture 6

## 1 Prony's Method (c. 1795)

**Definition 1.** *The Discrete Fourier Transform (DFT) - Consider* $\mathbf{x}$ *in* $\mathbb{R}^D$*. The DFT of* $\mathbf{x}$ *is:*

$$\hat{x}[k] = \frac{1}{\sqrt{D}} \sum_{n=0}^{D-1} x[n] e^{-2\pi i \frac{kn}{D}}$$

The DFT is a linear mapping. Let $\omega_D = \exp(-2\pi i/D)$ and define the matrix $\mathbf{F}$ with elements

$$F_{k,n} = \frac{e^{-2\pi i \frac{kn}{D}}}{\sqrt{D}} = \frac{\omega_D^{kn}}{\sqrt{D}}.$$

We can write the DFT as

$$\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

The matrix $\mathbf{F}$ is unitary, ($\mathbf{F}^\dagger \mathbf{F} = \mathbf{F}\mathbf{F}^\dagger = \mathbf{I}$). We can see this from direct evaluation:

$$\begin{aligned} [\mathbf{F}^\dagger \mathbf{F}]_{kk} &= \sum_{n=0}^{D} F_{nk} F_{nk} = \frac{1}{D} \sum_{n=0}^{D-1} \bar{\omega}^{nk} \omega^{nk} \\ [\mathbf{F}^\dagger \mathbf{F}]_{km} &= \frac{1}{D} \sum_{n=0}^{D-1} \bar{\omega}^{(m-k)n} = \frac{1}{D} \frac{(\omega^{m-k})^D - 1}{\omega^{m-k} - 1} = 0 \quad \text{for } m \neq k \end{aligned}$$

Since $\mathbf{F}$ is unitary, we immediately get the discrete analogs of the identities of Plancherel and Parseval. For $\mathbf{x}, \mathbf{y} \in \mathbb{C}^D$,

$$\begin{aligned} \mathbf{x}^\dagger \mathbf{y} &= \mathbf{F}\mathbf{x}^\dagger \mathbf{F}\mathbf{y} \qquad \textit{Plancherel} \\ \mathbf{x}^\dagger \mathbf{x} &= \mathbf{F}\mathbf{x}^\dagger \mathbf{F}\mathbf{x} \qquad \textit{Parseval}\,. \end{aligned}$$

We also have the following identity about convolutions. Recall that the convolution of $\mathbf{u}$ and $\mathbf{v} \in \mathbb{C}^D$ is defined as

$$\mathbf{u} * \mathbf{v}[k] = \sum_{m=0}^{D-1} u[m] v[(k-m) \mod D]\,.$$

Then we have the following

**Proposition 1.** *Let* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{C}^D$*. Let* $\mathbf{z} \in \mathbb{C}^D$ *be the vector with components* $z_k = x_k y_k$*. Then* $\hat{\mathbf{z}} = \hat{\mathbf{x}} * \hat{\mathbf{y}}$*.*

*Proof.*

$$\begin{aligned}
\hat{z}_k &= \frac{1}{\sqrt{D}} \sum_{n=0}^{D-1} e^{-2\pi i \frac{kn}{D}} z_n \\
&= \frac{1}{\sqrt{D}} \sum_{n=0}^{D-1} e^{-2\pi i \frac{kn}{D}} x_n y_n \\
&= \frac{1}{\sqrt{D}} \sum_{n=0}^{D-1} e^{-2\pi i \frac{kn}{D}} \left( \frac{1}{\sqrt{D}} \sum_{m=0}^{D-1} e^{2\pi i \frac{mn}{D}} \hat{x}_m \right) y_n \\
&= \frac{1}{\sqrt{D}} \sum_{m=0}^{D-1} \hat{x}_m \frac{1}{\sqrt{D}} \sum_{n=0}^{D-1} e^{-2\pi i \frac{(k-m)n}{D}} y_n \\
&= \frac{1}{\sqrt{D}} \sum_{m=0}^{D-1} \hat{x}[m] \hat{y}[(k-m) \mod D] = \hat{x} * \hat{y}[k]
\end{aligned}$$

□

We now can present Prony's Method which recovers an $s$-sparse vector from its first $2s$ Fourier coefficients

**Theorem 2** (Prony's Method)**.** *Let* $\mathbf{x}$ *be* $s$*-sparse. Let* $\hat{\mathbf{x}}$ *be its discrete Fourier Transform. Then* $\mathbf{x}$ *is uniquely determined by* $\hat{x}_0, \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{2S-1}$*. Moreover, an efficient algorithm for reconstructing* $\mathbf{x}$ *from these* $2s$ *numbers exists.*

*Proof.* Define a polynomial

$$p(z) = \prod_{k \in supp(x)} w_D^{-k}(-z + w_D^k) = 1 + \lambda_1 z + \lambda_2 z^2 + \ldots + \lambda_s z^s \,.$$

Then the degree of $p$ is and the roots are precisely $\{w_D^k\}_{k \in supp(x)}$

Define the vector $\mathbf{v} \in \mathbb{C}^D$ by

$$v = (1, \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_s, 0, \ldots, 0)^T$$

and set

$$\mathbf{u} = \mathbf{F}^\dagger \mathbf{v} = \frac{1}{\sqrt{D}} \sum_{k=0}^{D-1} w_D^{nk} v_k = \frac{1}{\sqrt{D}} \sum_{k=0}^{s} \lambda_k (w_D^n)^k = \frac{1}{\sqrt{D}} p(w_D^n)$$

By the above reasoning, we have $u_n = 0$ iff $n \in supp(x)$, and hence $q_n = x_n u_n = 0$ everywhere. It then follows from Proposition 1 that $0 = \hat{\mathbf{x}} * \mathbf{F}\mathbf{u} = \hat{\mathbf{x}} * \mathbf{v}$. Writing this last equation out in components, we see that

$$[\hat{\mathbf{x}} * \mathbf{v}]_k = \sum_{n=0}^{D-1} v_n \hat{x}[(k-n) \mod D] = \sum_{n=0}^{s} \lambda_n \hat{x}[(k-n) \mod D]$$

$$\begin{pmatrix} \hat{x}[k] \\ \hat{x}[(k-1) \mod D] \\ \hat{x}[(k-1) \mod D] \\ \dots \\ \hat{x}[(k-s) \mod D] \end{pmatrix}^T \begin{pmatrix} 1 \\ \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_s \end{pmatrix} = 0 \text{ for any } k$$

In other words, substituting $k = t + s(t < D - s)$,

$$\begin{pmatrix} \hat{x}_{t+s} \\ \hat{x}_{t+s-1} \\ \hat{\dots} \\ \hat{x}_t \end{pmatrix}^T \begin{pmatrix} 1 \\ \lambda_1 \\ \dots \\ \lambda_s \end{pmatrix}$$

Using $t = 0, \dots, s-1$, we can construct $s$ equations with $s$ unknowns

$$\begin{pmatrix} \hat{x}_0 & \hat{x}_1 & \hat{x}_2 & \dots & \hat{x}_{s-1} \\ \hat{x}_1 & \hat{x}_2 & \hat{x}_3 & \dots & \hat{x}_s \\ \hat{x}_2 & \hat{x}_3 & \hat{x}_4 & \dots & \hat{x}_{s+1} \\ \dots \\ \hat{x}_{s-1} & \hat{x}_s & \hat{x}_{s+1} & \dots & \hat{x}_{2s-2} \end{pmatrix} \begin{pmatrix} \lambda_s \\ \lambda_{s-1} \\ \lambda_{s-2} \\ \dots \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} \hat{x}_s \\ \hat{x}_{s+1} \\ \hat{x}_{s+2} \\ \dots \\ \hat{x}_{2s-1} \end{pmatrix} \tag{1}$$

This system has full rank if $\mathbf{x}$ is $s$-sparse (see Badri's notes). Thus, we can compute the coefficients of $p$ which uniquely determine the support of $\mathbf{x}$. From the support, we can then solve for the non-zero coefficients of $\mathbf{x}$. $\square$

Prony's method is thus summarized by the following algorithm

### Algorithm

- Form and solve the linear system (1) for $\lambda$
- Compute the roots of $1 + \lambda_1 z + \dots + \lambda_s z^s$ to find $supp(x)$
- solve the linear system $\{\hat{x}_t = \sum_{k \in supp(x)} F_{kn} w_n\}_{k=0}^{2s-1}$ for $w$
- Return

$$x_k = \begin{cases} w_k & k \in supp(x) \\ 0 & \text{otherwise} \end{cases}$$

Prony's method shows that there is a deterministic $2s \times n$ matrix that can encode all $s$-sparse vectors, and that any $s$-sparse vector can be decoded from its image. Namely, this matrix is $\mathbf{F}^{(2s)}$, the matrix with the first $2s$ rows of $\mathbf{F}$. It's not hard to see that $2s$ is the minimal possible size for any such linear encoding. Let $\mathbf{A}$ be a $p \times D$ matrix with $p < 2s$. Let $\tilde{\mathbf{A}}$ be the submatrix of the first $2s$ columns. Then there must exist a $\mathbf{w} \in \mathbb{R}^{2s}$ such that $\tilde{\mathbf{A}}\mathbf{w} = 0$. Define the vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$ by

$$\begin{aligned} \mathbf{x} &= (w_1, \quad \dots, \quad w_s, \quad 0, \dots, 0) \\ \mathbf{y} &= (0, \quad \dots, \quad 0, \quad -w_{s+1}, \dots, -w_{2s}, 0, \dots, 0) \end{aligned}$$

Then $\mathbf{Ax} = \mathbf{Ay}$, and hence 2 $s$-sparse vectors are mapped to the same point by $\mathbf{A}$.

The main drawbacks of Prony's method is that it is not numerically stable for large $D$. Moreover, it is not stable if $\mathbf{x}$ is nearly sparse. Also, the algorithm is very specific to the structure of the Fourier transform and does not generalize to other linear encodings. In the next lecture, we explore an algorithm that overcomes all of these shortcomings, at the expense of requiring a slightly larger encoding matrix $\mathbf{A}$.