Scaling frames via linear programming

Recall, a frame $\Phi = \{\varphi_k\}_{k=1}^M \subseteq \mathbb{R}^N$ is *scalable* if there exists an $M \times M$ diagonal matrix $X = \mathrm{diag}(x_k)$ such that

$$\tilde{S} = \Phi X^T X \Phi^T = \Phi X^2 \Phi^T = AI$$

for some constant $A > 0$. We can convert this into a linear system of equations in M unknowns $\{x_k^2\}$.

Define $F : \mathbb{R}^N \to \mathbb{R}^d$ as

$$F(x) = [F_0(x), F_1(x), \dots, F_{N-1}(x)]^T,$$

where

$$F_0(x) = \begin{bmatrix} x_1^2 - x_2^2 \\ x_1^2 - x_3^2 \\ \vdots \\ x_1^2 - x_N^2 \end{bmatrix}, \quad F_k(x) = \begin{bmatrix} x_k x_{k+1} \\ x_k x_{k+2} \\ \vdots \\ x_k x_N \end{bmatrix}.$$

Here, $F_0(x) \in \mathbb{R}^{N-1}$ and $F_k(x) \in \mathbb{R}^{N-k}$ for $k = 1, \dots, N-1$ and $d = \frac{(N-1)(N+2)}{2}$. We will make use of the $d \times M$ matrix

$$F(\Phi) \triangleq (F(\varphi_1), F(\varphi_2), \dots, F(\varphi_M)).$$

In particular, Φ is scalable if and only if there exists a non-negative, non-zero vector in $\ker F(\Phi)$. We can attempt to find such a vector using linear programming.

For some coefficient vector $a \in \mathbb{R}^M$ with no zero components, define a linear program $\mathcal{P}$ as

$$\begin{aligned} (\mathcal{P}) \text{ minimize: } & a^T u \\ \text{subject to: } & F(\Phi)u = 0 \\ & ||u||_1 = 1 \\ & u \geq 0 \end{aligned}$$

Any solution u to this linear program is associated to a scaling matrix X_u whose diagonal elements are given by $X_{ii} = \sqrt{u_i}$, and $X_u\Phi$ will be a tight frame. Any a may be chosen; common choices have $a_i = 1$ for all i, or such that $a_i = \frac{1}{||F(\varphi_i)||_2}$.

Numerical exercises

Please visit the URL provided for MATLAB files to solve this linear programming problem, as well as auxiliary files for generating random Gaussian and Bernoulli frames.

(1) Let $N = 4$. For each $M = 6, 10, 20, 30, 40, 50, 60$, generate several (10-50) random frames Φ of M elements for $\mathbb{R}^N$. For each such frame, call scale_linear_program.m with the frame as input.

(2) For each value of M, what percentage of the randomly-generated frames are scalable? Is there a transition point at which this percentage changes substantially?

(3) For $N = 5, 6, 7, 8$, repeat this exercise with a number of M values between N and $2N^2$ as computational power allows. What percentage of randomly-generated frames are scalable? What is the relationship between N and any scalability transition values M you observe?