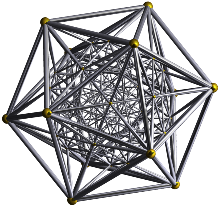
Computational Principles for High-dim Data Analysis

(Lecture Three)

Yi Ma

EEECS Department, UC Berkeley

September 2, 2021



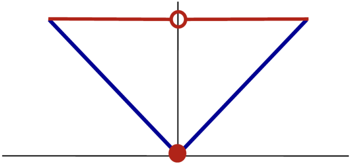
Relaxing the Sparse Recovery Problem

Why Convexification?

Intuitive reasons why ℓ^0 minimization:

$$\min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{1}$$

is very challenging:



Not amenable to local search methods such as gradient descent.

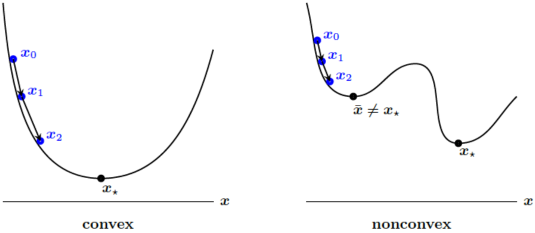
Convex versus Nonconvex Functions

For minimizing a generic function:

$$\min f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathsf{C} \text{ (a convex set)}, \tag{2}$$

conduct **local gradient descent search:** (Appendix D)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k). \tag{3}$$



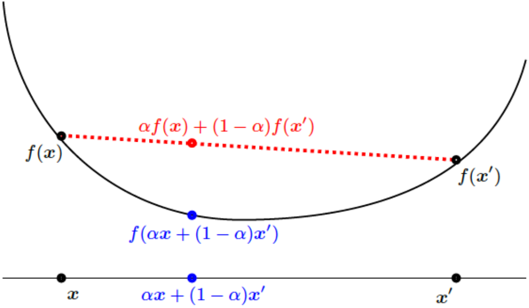
Intuitively, **convexity lends to global optimality.**

Convex Functions [Appendix B]

Definition (Convex Function)

A continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for every pair of points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ and $\alpha \in [0, 1]$ it satisfies the Jensen's inequality:

$$f\Big(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}'\Big) \;\le\; \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}'). \tag{4}$$



Global Optimality

Proposition

Any local minimum of a convex function is also a global minimum.

Proof.

Let $\bar{\boldsymbol{x}}$ be a local minimum: $\forall \boldsymbol{x} : \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2 \leq \epsilon$, we have $f(\bar{\boldsymbol{x}}) \leq f(\boldsymbol{x})$.
Assume $\boldsymbol{x}_\star$ is the global minimum and $f(\bar{\boldsymbol{x}}) > f(\boldsymbol{x}_\star)$.
Choose λ such that $\boldsymbol{x}_\lambda = \lambda\bar{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}_\star$ satisfies $\|\boldsymbol{x}_\lambda - \bar{\boldsymbol{x}}\|_2 \leq \epsilon$. Then

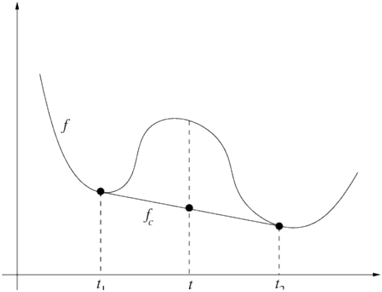
$$\begin{aligned} f(\bar{\boldsymbol{x}}) &\leq f(\boldsymbol{x}_\lambda) \\ &\leq f(\lambda\bar{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}_\star) \\ &\leq \lambda f(\bar{\boldsymbol{x}}) + (1-\lambda) f(\boldsymbol{x}_\star) \\ &< f(\bar{\boldsymbol{x}}). \end{aligned}$$

□

Convex Envelope

Definition (Lower Convex Envelope)

A function $f_c(\boldsymbol{x})$ is said to be a (lower) **convex envelope** of $f(\boldsymbol{x})$ if for all convex functions $g \leq f$ we have $g \leq f_c$.



Lower convex envelope f_c is well and uniquely defined and is equivalent to the **convex biconjugate** function f^{**} of f.

The ℓ^1 Norm as Envelope of ℓ^0 Norm

$$\forall \boldsymbol{x} \in \mathbb{R}^n : \quad \|\boldsymbol{x}\|_0 = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}(i) \neq 0}, \quad \|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |\boldsymbol{x}(i)|. \tag{5}$$

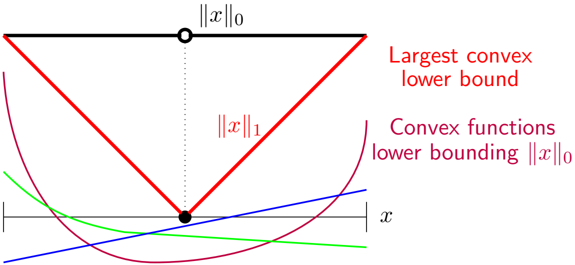


Figure: **Convex surrogates for the ℓ^0 norm.** $|x|$ is the *convex envelope* of $\|x\|_0$ on $[-1, 1]$.

The ℓ^1 Norm as Envelope of ℓ^0 Norm

Theorem

The function $\|\cdot\|_1$ is the convex envelope of $\|\cdot\|_0$, over the set $\mathsf{B}_\infty = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_\infty \leq 1\}$ of vectors whose elements all have magnitude at most one.

Proof.

Consider the cube $\mathsf{C} = [0,1]^n$ with vertex vectors $\boldsymbol{\sigma} \in \{0,1\}^n$. For any convex function $f \leq \|\cdot\|_0$,

$$
\begin{aligned}
f(\boldsymbol{x}) &= f\Big(\sum_i \lambda_i \boldsymbol{\sigma}_i\Big) \leq \sum_i \lambda_i f(\boldsymbol{\sigma}_i) && \text{[Jensen's inequality]} \\
&\leq \sum_i \lambda_i \|\boldsymbol{\sigma}_i\|_0 = \sum_i \lambda_i \|\boldsymbol{\sigma}_i\|_1 && [\boldsymbol{\sigma}_i \text{ are binary}] \\
&= \|\boldsymbol{x}\|_1 . && (6)
\end{aligned}
$$

Repeat the argument for each orthants. □

Sparsity Promoting Property of Norms

A Toy Problem: given a vector

$$\vec{\boldsymbol{v}}(t) = [t, t-1, t-1]^* \quad \in \mathbb{R}^3,$$

find t such that $\vec{\boldsymbol{v}}$ is sparse.

Strategy: given a certain norm $\|\cdot\|$,

$$\min_t f(t) = \|\vec{\boldsymbol{v}}(t)\|.$$

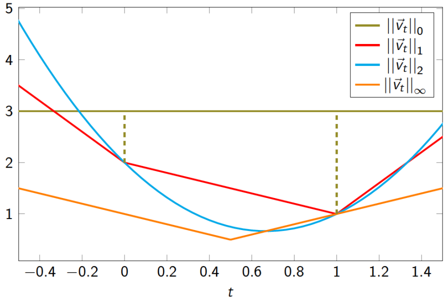


Figure courtesy of Carlos Fernandez of NYU.

Minimizing the ℓ^1 Norm

Replace ℓ^0 minimization:

$$\min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{7}$$

with the relaxed ℓ^1 minimization:

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{8}$$

Two technical difficulties:

- **Nontrivial constraints:** Unlike the general unconstrained problem (2), in the problem (8) the solution $\boldsymbol{x}$ must satisfy $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.
- **Nondifferentiable objective:** ℓ^1 norm in (8) is not differentiable. So around points of interest the gradient $\nabla f(\boldsymbol{x})$ does not exist.

ℓ^1 Minimization via Linear Programming

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{9}$$

Let

$$\boldsymbol{x}^+ = \max\{\boldsymbol{x}, \boldsymbol{0}\}, \quad \text{and} \quad \boldsymbol{x}^- = \max\{-\boldsymbol{x}, \boldsymbol{0}\}.$$

Let $\boldsymbol{z} = \left[\begin{smallmatrix} \boldsymbol{x}^+ \\ \boldsymbol{x}^- \end{smallmatrix}\right] \in \mathbb{R}^{2n}$ and we have:

$$\|\boldsymbol{x}\|_1 = \boldsymbol{1}^*(\boldsymbol{x}^+ + \boldsymbol{x}^-) = \boldsymbol{1}^*\boldsymbol{z} \quad \text{and} \quad \boldsymbol{A}\boldsymbol{x} = [\boldsymbol{A}, -\boldsymbol{A}]\boldsymbol{z}. \tag{10}$$

Then ℓ^1 minimization is equivalent to an LP problem:

$$\min_{\boldsymbol{z}} \boldsymbol{1}^*\boldsymbol{z} \quad \text{subject to} \quad [\boldsymbol{A}, -\boldsymbol{A}]\boldsymbol{z} = \boldsymbol{y},\ \boldsymbol{z} \geq \boldsymbol{0}. \tag{11}$$

This LP problem can be solved in polynomial time.

Minimizing the ℓ^1 Norm via Local Greedy Descent

For minimizing a function with **constraints** (Appendix C& D):

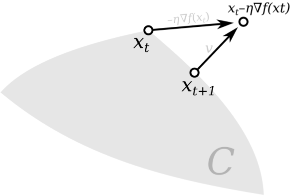
$$\min f(\boldsymbol{x}), \quad \text{subject to} \quad \boldsymbol{x} \in \mathsf{C} \text{ (a convex set)}, \tag{12}$$

Basic Strategy: projected gradient descent (PGD):

$$\boldsymbol{x}_{k+1} = \mathcal{P}_{\mathsf{C}}\left[\, \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) \,\right]. \tag{13}$$

where $\mathcal{P}_{\mathsf{C}}$ projects
a point, say $\boldsymbol{z}$, to the nearest point in C:

$$\mathcal{P}_{\mathsf{C}}[\boldsymbol{z}] = \arg\min_{\boldsymbol{x} \in \mathsf{C}} \ \tfrac{1}{2} \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \equiv h(\boldsymbol{x}). \tag{14}$$

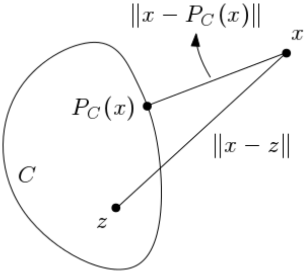


Projection on a Convex Set

How to find the nearest point $\hat{\boldsymbol{x}} = \mathcal{P}_{\mathsf{C}}[\boldsymbol{x}]$ to a point $\boldsymbol{x}$ in a set $\mathsf{C} = \{\boldsymbol{z} \mid h(\boldsymbol{z}) \le c\}$?

Fact: $\hat{\boldsymbol{x}}$ satisfies two conditions:

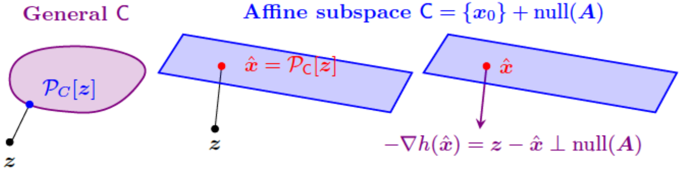
1. Feasibility: $h(\hat{\boldsymbol{x}}) \le c$;
2. Optimality:
   $-\nabla h(\hat{\boldsymbol{x}})$ is orthogonal to C at $\hat{\boldsymbol{x}}$.



Project onto a flat: $\mathsf{C} = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$

In this special case, $\hat{\boldsymbol{x}}$ satisfies two conditions:

1. Feasibility: $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$;
2. Optimality: $\boldsymbol{z} - \hat{\boldsymbol{x}} \perp \mathrm{null}(\boldsymbol{A})$.



From these conditions, we have:

$$\hat{\boldsymbol{x}} = \mathcal{P}_{\{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}}[\boldsymbol{z}] = \boldsymbol{z} - \boldsymbol{A}^* \left(\boldsymbol{A}\boldsymbol{A}^*\right)^{-1} [\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}] . \tag{15}$$

Directly check? Or derive alternatively? (exercise 2.11)

Minimizing the ℓ^1 Norm: Nondifferentiability

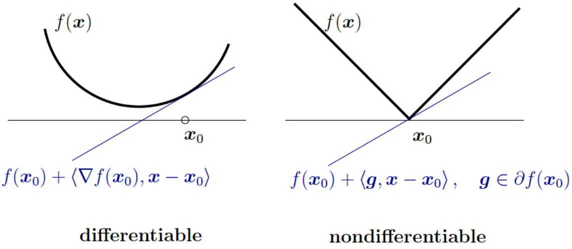
Try to solve:

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{16}$$

using projected gradient descent:

$$\min f(\boldsymbol{x}): \quad \boldsymbol{x}_{k+1} = \mathcal{P}_{\mathsf{C}}\left[\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)\right]. \tag{17}$$

But $\|\boldsymbol{x}\|_1$ is not differentialble.



Design Strategies for All Local Descent Methods

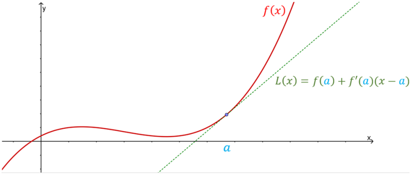
Minimization via local descent (Appendix D):

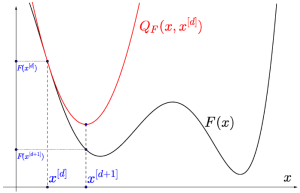
$$\begin{array}{rccc} \min f(\boldsymbol{x}): & \boldsymbol{x}_k & \rightarrow & \boldsymbol{x}_{k+1} \\ \text{such that} & f(\boldsymbol{x}_k) & \geq & f(\boldsymbol{x}_{k+1}). \end{array}$$

At current iterate $\boldsymbol{x}_k$, find **a local surrogate** $\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \approx f(\boldsymbol{x})$ such that

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x} \in \mathsf{C}} \hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \quad \text{easy to find!} \tag{18}$$

where $\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k)$ could be linear, quadratic, higher-order; or upper-bound (conservative) or lower-bound (accelerating).





Subgradient and Subdifferential

Generalizing the gradient $\nabla f(\boldsymbol{x})$ at $\boldsymbol{x}_0$ with the property:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n. \tag{19}$$

Definition (Subgradient and Subdifferential)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. A *subgradient* of f at $\boldsymbol{x}_0$ is any vector $\boldsymbol{u} \in \mathbb{R}^n$ satisfying

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle, \quad \forall\, \boldsymbol{x}. \tag{20}$$

The *subdifferential* of f at $\boldsymbol{x}_0$ is the set of all subgradients of f at $\boldsymbol{x}_0$:

$$\partial f(\boldsymbol{x}_0) = \{\boldsymbol{u} \mid \forall\, \boldsymbol{x} \in \mathbb{R}^n,\ f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle\}. \tag{21}$$

Subgradient and Subdifferential of ℓ^1 Norm

Lemma (Subdifferential of $\|\cdot\|_1$)

Let $\boldsymbol{x} \in \mathbb{R}^n$, with $\mathsf{I} = supp(\boldsymbol{x})$,

$$\partial \|\cdot\|_1 (\boldsymbol{x}) = \{\boldsymbol{v} \in \mathbb{R}^n \mid \boldsymbol{P}_{\mathsf{I}}\boldsymbol{v} = \operatorname{sign}(\boldsymbol{x}),\ \|\boldsymbol{v}\|_\infty \le 1\}. \tag{22}$$

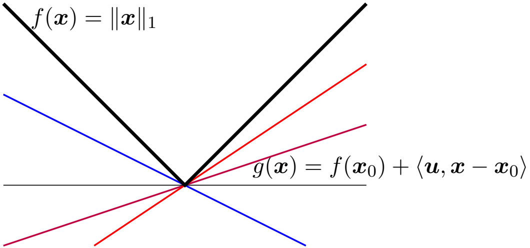


Figure: In **blue**, **purple**, and **red**, three linear lower bounds, taken at $\boldsymbol{x}_0 = \mathbf{0}$, with slope $\boldsymbol{u} = -\frac{1}{2}$, $\frac{1}{3}$, and $\frac{2}{3}$, respectively. Any slope $\boldsymbol{u} \in [-1, 1]$ defines a linear lower bound on $f(\boldsymbol{x})$ around $\boldsymbol{x}_0 = \mathbf{0}$. So, $\partial|\cdot|(0) = [-1, 1]$. For $\boldsymbol{x}_0 > 0$, the only linear lower bound has slope $\boldsymbol{u} = 1$; for $\boldsymbol{x}_0 < 0$, the only linear lower bound has slope $\boldsymbol{u} = -1$. So, $\partial|\cdot|(\boldsymbol{x}) = \{-1\}$ for $\boldsymbol{x} < 0$ and $\partial|\cdot|(\boldsymbol{x}) = \{1\}$ for $\boldsymbol{x} > 0$.

Minimizing the ℓ^1 Norm: Projected Subgradient

To solve:

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{23}$$

using projected subgradient descent:

$$\boldsymbol{x}_{k+1} = \mathcal{P}_{\mathsf{C}}[\boldsymbol{x}_k - t_k \boldsymbol{g}_k], \quad \boldsymbol{g}_k \in \partial f(\boldsymbol{x}_k). \tag{24}$$

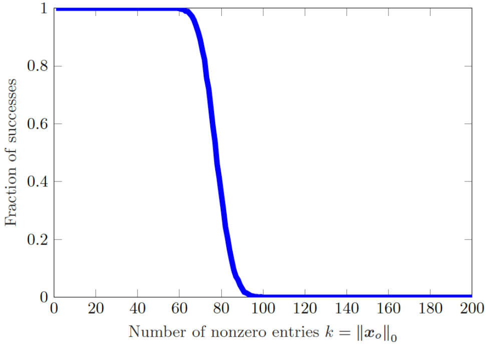
Algorithm (ℓ^1 Minimization via Projected Subgradient Descent):

1: **Input:** a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{y} \in \mathbb{R}^m$.
2: Compute $\boldsymbol{\Gamma} \leftarrow \boldsymbol{I} - \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}\boldsymbol{A}$, and $\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{A}^\dagger \boldsymbol{y} = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}\boldsymbol{y}$.
3: $\boldsymbol{x}_0 \leftarrow \boldsymbol{0}$.
4: $t \leftarrow 0$.
5: **repeat many times**
6: $\quad t \leftarrow t + 1;$
7: $\quad \boldsymbol{x}_t \leftarrow \tilde{\boldsymbol{x}} + \boldsymbol{\Gamma}\left(\, \boldsymbol{x}_{t-1} - \frac{1}{t}\operatorname{sign}(\boldsymbol{x}_{t-1}) \,\right);$
8: **end while**

Minimizing the ℓ^1 Norm: Simulations

$$\textbf{Solve:} \quad \min \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{25}$$

$\boldsymbol{A}$ is of size 200×400. Fraction of success across 50 trials.



Error Correction via ℓ^1 Minimization

Let $\boldsymbol{F} \in \mathbb{C}^{n \times n}$ be the **Discrete Fourier Transform** (DFT), and $\boldsymbol{B} \in \mathbb{C}^{n \times (d+1)}$ be a submatrix of the d lowest-frequency elements of this basis and their conjugates:

$$\boldsymbol{B} = \left[\boldsymbol{f}_{-\frac{d-1}{2}} \mid \cdots \mid \boldsymbol{f}_{\frac{d-1}{2}} \right] \quad \in \mathbb{C}^{n \times (d+1)}, \tag{26}$$

$$\boldsymbol{y} = \boldsymbol{x}_o + \boldsymbol{e}_o, \quad \text{where } \ \boldsymbol{x}_o = \boldsymbol{B}\boldsymbol{w}_o \ \text{ and } \ \|\boldsymbol{e}_o\|_0 \leq k. \tag{27}$$

Discrete Logan's Theorem:

$$\min \|\boldsymbol{y} - \boldsymbol{x}\|_1 \quad \text{s.t.} \quad \boldsymbol{x} \in \operatorname{col}(\boldsymbol{B}). \tag{28}$$

Error Correction via ℓ^1 Minimization

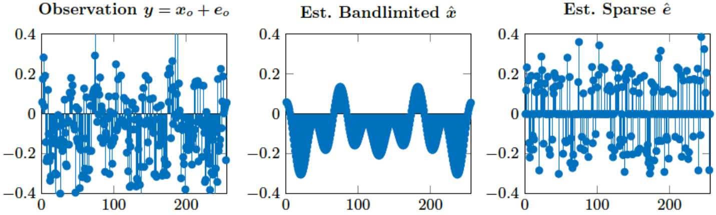
Let $\boldsymbol{A}$ be the (left) orthogonal complement to $\boldsymbol{B}$: $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0}$. Then:

$$\bar{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}_o + \boldsymbol{e}_o) = \boldsymbol{A}\boldsymbol{e}_o. \tag{29}$$

To solve for $\boldsymbol{e}_o$:

$$\min \|\boldsymbol{e}\|_1 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{e} = \bar{\boldsymbol{y}}. \tag{30}$$

According to Logan's Theorem, this succeeds if $d \times k \leq c\frac{\pi}{2}$.



What about other frequency components of $\boldsymbol{F}$?

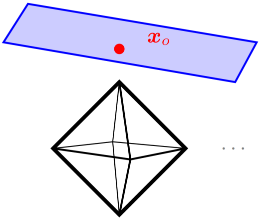
Next: Towards a Rigorous Justification

Given $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ with $\boldsymbol{x}_o$ sparse:

$$\textbf{NP:} \quad \min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{31}$$

$$\textbf{P:} \quad \min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{32}$$

When and Why does ℓ^1 minimization work?



Assignments

- Reading: Section 2.3 of Chapter 2.
- Reading: Appendix C & D.
- Programming Homework # 1.