Optimization with Sparsity Inducing Terms

Amir Beck

School of Mathematical Sciences, Tel-Aviv University

Based on joint works with Nadav Hallak

Optimization and Discrete Geometry: Theory and Practice, April 24-47, 2018, Tel Aviv University

Optimization Problems with an ℓ_0-"norm" I

ℓ_0-"norm":

$$\|\mathbf{x}\|_0 = \#\{i : x_i \neq 0\}$$

nonconvex, noncontinuous, but at least closed...

$$\|(-1,2,0,0)^T\|_0 = 2, \|(0,0,0,10)^T\|_0 = 1.$$

Optimization Problems with an ℓ_0-"norm" I

ℓ_0- "norm":

$$\|\mathbf{x}\|_0 = \#\{i : x_i \neq 0\}$$

nonconvex, noncontinuous, but at least closed...

$$\|(-1,2,0,0)^T\|_0 = 2, \|(0,0,0,10)^T\|_0 = 1.$$

- **Sparsity-Constrained Problems**

$$(C) \quad \begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in C_s \cap B, \end{array}$$

where $C_s = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_0 \leq s\}$

Difficulties:

(a) $C_s \cap B$ non-convex

(b) $C_s \cap B$ induces a combinatorial constraint

No global optimality conditions, "solution" methods are heuristic

Optimization Problems with an ℓ_0-"norm" II

- **Sparsity-Penalized Problems** $(\lambda > 0)$

$$(C) \quad \begin{array}{ll} \min & f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 \\ \text{s.t.} & \mathbf{x} \in B. \end{array}$$

As opposed to convex programming, the penalized and constrained problems are not equivalent.

Examples

- **(Linear) Compressed Sensing.** Recover a sparse signal $\mathbf{x}$ with a sampling matrix $\mathbf{A}$ and a measure $\mathbf{b}$.

$$(CS) \quad \begin{array}{ll} \min & \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in C_s \cap \mathbb{R}^n \end{array} \qquad \text{or} \qquad \min \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_0$$

Examples

- **(Linear) Compressed Sensing.** Recover a sparse signal $\mathbf{x}$ with a sampling matrix $\mathbf{A}$ and a measure $\mathbf{b}$.

$$(CS) \quad \begin{array}{ll} \min & \|\mathbf{Ax}-\mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in C_s \cap \mathbb{R}^n \end{array} \quad \text{or} \quad \min \|\mathbf{Ax}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_0$$

- **Sparse Index Tracking.** Track an index $\mathbf{b}$ with a few assets, with return matrix $\mathbf{A}$.

$$(IT) \quad \begin{array}{ll} \min & \|\mathbf{Ax}-\mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in C_s \cap \Delta_n \end{array} \quad \textit{or} \quad \min\{\|\mathbf{Ax}-\mathbf{b}\|_2^2+\lambda\|\mathbf{x}\|_0 : \mathbf{x} \in \Delta_n\}$$

$(\Delta_n = \{\mathbf{x} : \mathbf{e}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\})$ (Takeda et al '12)

Examples

- **(Linear) Compressed Sensing.** Recover a sparse signal $\mathbf{x}$ with a sampling matrix $\mathbf{A}$ and a measure $\mathbf{b}$.

$$(CS) \quad \begin{array}{ll} \min & \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in C_s \cap \mathbb{R}^n \end{array} \quad \text{or} \quad \min \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_0$$

- **Sparse Index Tracking.** Track an index $\mathbf{b}$ with a few assets, with return matrix $\mathbf{A}$.

$$(IT) \quad \begin{array}{ll} \min & \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \\ \text{s.t.} & \mathbf{x} \in C_s \cap \Delta_n \end{array} \quad \textit{or} \quad \min\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\lambda\|\mathbf{x}\|_0 : \mathbf{x} \in \Delta_n\}$$

$(\Delta_n = \{\mathbf{x} : \mathbf{e}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\})$ (Takeda et al '12)

- **Sparse Principal Component Analysis** Find the dominant sparse principal eigenvector of a matrix $\mathbf{A}$.

$$(PCA) \quad \begin{array}{ll} \max & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{s.t.} & \mathbf{x} \in C_s \cap B_2[\mathbf{0}, 1] \end{array} \quad \textit{or} \quad \max\{\mathbf{x}^T\mathbf{A}\mathbf{x}-\lambda\|\mathbf{x}\|_0 : \mathbf{x} \in B_2[0,1]\}$$

Moghaddam, Weiss, Avidan '06, d'Aspremont, Bach, El-Ghaoui '08, d'Aspremont, El-Ghaoui, Jordan, Lanckriet '07, Luss and Teboulle '13

Literature - CS

- Linear:
  1. **Conditions for reconstruction:** RIP (Candes and Tao '05), SRIP (Beck and Teboulle '10), spark (Donoho and Elad '03; Gorodnitsky and Rao '97), mutual coherence (Donoho et al. '03; Donoho and Huo '99; Mallat and Zhang '93)
  2. **Reviews:** Bruckstein et al. '09, Davenport et al. '11, Tropp and Wright '10.
  3. **Iterative algorithms:** IHT (Blumensath and Davis '08, '09, '12; Beck and Teboulle '10), CoSaMP (Needell and Tropp '09)
- Nonlinear:
  1. **Phase retrieval:** Shechtman et al. '13; Ohlsson and Eldar '13; Eldar and Mendelson '13; Eldar et al. '13; Hurt. '89
  2. **Nonlinear:** optimality conditions (Beck and Eldar '13), GraSP (Bahmani et al. '13)

Objectives

Unifying the first two models:

The sparse optimization model

$$(P) \quad \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

where either $g(\mathbf{x}) = g_1(\mathbf{x}) \equiv \delta_{B \cap C_s}(\mathbf{x})$ (model 1) or $g(\mathbf{x}) = g_2(\mathbf{x}) \equiv \lambda \|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$ (model 2)

B is a nonempty closed and convex set. $\delta_C(\mathbf{x}) = 0$ for $\mathbf{x} \in C$ and ∞ for $\mathbf{x} \notin C$.

Objectives

Unifying the first two models:

The sparse optimization model

$$(P) \quad \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

where either $g(\mathbf{x}) = g_1(\mathbf{x}) \equiv \delta_{B \cap C_s}(\mathbf{x})$ (model 1) or $g(\mathbf{x}) = g_2(\mathbf{x}) \equiv \lambda \|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$ (model 2)

B is a nonempty closed and convex set. $\delta_C(\mathbf{x}) = 0$ for $\mathbf{x} \in C$ and ∞ for $\mathbf{x} \notin C$.

Main Objectives:

- Define necessary **optimality conditions**
- Develop corresponding **algorithms**
- Establish **hierarchy** between algorithms and conditions

The case $B = \mathbb{R}^n$: Beck, Eldar '13

Objectives

Unifying the first two models:

The sparse optimization model

$$(P) \quad \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

where either $g(\mathbf{x}) = g_1(\mathbf{x}) \equiv \delta_{B \cap C_s}(\mathbf{x})$ (model 1) or $g(\mathbf{x}) = g_2(\mathbf{x}) \equiv \lambda \|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$ (model 2)

B is a nonempty closed and convex set. $\delta_C(\mathbf{x}) = 0$ for $\mathbf{x} \in C$ and ∞ for $\mathbf{x} \notin C$.

Main Objectives:

- Define necessary **optimality conditions**
- Develop corresponding **algorithms**
- Establish **hierarchy** between algorithms and conditions

The case $B = \mathbb{R}^n$: Beck, Eldar '13

However, we will also need to study and compute Proximal Mappings of g_1 and g_2.

Recap of Necessary First Order Opt. for the Composite Model with (some) Convexity: Stationarity

$$(*) \ \min\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x})\}$$

f continuously differentiable (not necessarily convex), g proper, closed and convex.

Recap of Necessary First Order Opt. for the Composite Model with (some) Convexity: Stationarity

$$(*) \ \min\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x})\}$$

f continuously differentiable (not necessarily convex), g proper, closed and convex.

Equivalent Definitions of Stationarity: $\mathbf{x}^*$ stationary point iff

Prox Form: for some $L > 0$

$$\mathbf{x}^* = \mathrm{prox}_{\frac{1}{L}g}\left(\mathbf{x}^* - \frac{1}{L}\nabla f(\mathbf{x}^*)\right)$$

Variational Form

$$F'(\mathbf{x}^*, \mathbf{y} - \mathbf{x}^*) \geq 0 \forall \mathbf{y} \in \operatorname{dom} g$$

Recap of Necessary First Order Opt. for the Composite Model with (some) Convexity: Stationarity

$$(*) \ \min\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x})\}$$

f continuously differentiable (not necessarily convex), g proper, closed and convex.

Equivalent Definitions of Stationarity: $\mathbf{x}^*$ stationary point iff

Prox Form: for some $L > 0$

$$\mathbf{x}^* = \mathrm{prox}_{\frac{1}{L}g}\left(\mathbf{x}^* - \frac{1}{L}\nabla f(\mathbf{x}^*)\right)$$

Variational Form

$$F'(\mathbf{x}^*, \mathbf{y} - \mathbf{x}^*) \geq 0 \forall \mathbf{y} \in \mathrm{dom}\, g$$

- conditions are equivalent $\Rightarrow$ independent of L
- most 1st order algorithms converge to stat. points.
- condition relies on the properties/computability of $\mathrm{prox}_g(\cdot)$

$$\mathrm{prox}_g(\mathbf{x}) = \underset{\mathbf{y}}{\mathrm{argmin}}\left\{g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2\right\}.$$

Why Study Proximal Mappings?

$$\mathrm{prox}_g(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}$$

To define optimality conditions, we need to

- compute and analyze properties of $\mathrm{prox}_{g_1}, \mathrm{prox}_{g_2}$.

Computing $\mathrm{prox}_{g_1}, \mathrm{prox}_{g_2}$ is in general a difficult task, but in fact tractable under assumptions such as symmetry of B

Why Study Proximal Mappings?

$$\mathrm{prox}_g(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}$$

To define optimality conditions, we need to

- compute and analyze properties of $\mathrm{prox}_{g_1}, \mathrm{prox}_{g_2}$.

Computing $\mathrm{prox}_{g_1}, \mathrm{prox}_{g_2}$ is in general a difficult task, but in fact tractable under assumptions such as symmetry of B

Revised Layout:

Proximal Mappings, Optimality Conditions, Algorithms

Proximal Mappings of g_1 and g_2

Proximal Mapping of g_1

Sparse projection over B:

$$\mathrm{prox}_{g_1}(\mathbf{x}) = P_{B \cap C_s}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 : \mathbf{y} \in B \cap C_s \right\}$$

- proximal mapping=orthogonal projection onto $B \cap C_s$.

Proximal Mapping of g_1

Sparse projection over B:

$$\mathrm{prox}_{g_1}(\mathbf{x}) = P_{B\cap C_s}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 : \mathbf{y} \in B \cap C_s \right\}$$

- proximal mapping=orthogonal projection onto $B \cap C_s$.
- If $B = \mathbb{R}^n$, then $P_{C_s \cap B}(\mathbf{x}) = P_{C_s}(\mathbf{x})$ comprises all vectors consisting of the s components of $\mathbf{x}$ with the largest absolute values and with zeros elsewhere.
- In general, a multi-valued mapping.

Supports, Super Supports

Let $\mathbf{x} \in \mathbb{R}^n$, $s \in [n] = \{1, \ldots, n\}$.

1. **Support** of $\mathbf{x}$: $I_1(\mathbf{x}) \equiv \{i \in [n] \ : \ x_i \neq 0\}$.
2. **Super support** of $\mathbf{x}$: any set T s.t. $I_1(\mathbf{x}) \subseteq T$ and $|T| = s$.
3. $\mathbf{x}$ has **full support** if $\|\mathbf{x}\|_0 = |I_1(\mathbf{x})| = s$.
4. **Off-support** of $\mathbf{x}$: $I_0(\mathbf{x}) \equiv \{i \in [n] \ : \ x_i = 0\}$.

Example

$s = 3, n = 5$ and $\mathbf{x} = (-3, 4, 0, 0, 0)^T$

1. **Support**: $I_1(\mathbf{x}) = \{1, 2\}$
2. **Super support**: $T \in \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}\}$
3. **Incomplete support**: $\|\mathbf{x}\|_0 < s$
4. **Off-support**: $I_0(\mathbf{x}) = \{3, 4, 5\}$

Restriction to Index Sets

$\mathbf{x} \in \mathbb{R}^n$, $T \subseteq [n]$ index set

1. $\mathbf{x}_T \in \mathbb{R}^{|T|}$ is the restriction of $\mathbf{x}$ to T
2. $B_T = \{\mathbf{x} \in \mathbb{R}^{|T|} : \mathbf{U}_T\mathbf{x} \in B\}$ is **the restriction of B to T**

Example

$$\mathbf{x} = (8,7,6,5)^T \Rightarrow \mathbf{x}_{1,3} = (8,6)^T.$$

$$B = \{(x_1, x_2, x_3, x_4) : x_1 + 2x_2 + 3x_3 + 4x_4 = 1\}$$
$$\Downarrow$$
$$B_{1,2} = \{(x_1, x_2)^T : x_1 + 2x_2 = 1\}$$

Phases in Computing the Projection

To find $\mathbf{y} \in P_{C_s \cap B}(\mathbf{x})$:

(1) find its super support S

(2) Compute $\mathbf{y}_S = P_{B_S}(\mathbf{x}_S), \ \mathbf{y}_{S^c} = \mathbf{0}$

- **Naive approach:** go over all possible $\binom{n}{s}$ super supports, compute the corresponding projections, and find the sparse projection vector. TOO EXPENSIVE.

Phases in Computing the Projection

To find $\mathbf{y} \in P_{C_s \cap B}(\mathbf{x})$:

(1) find its super support S

(2) Compute $\mathbf{y}_S = P_{B_S}(\mathbf{x}_S), \ \mathbf{y}_{S^c} = \mathbf{0}$

- **Naive approach:** go over all possible $\binom{n}{s}$ super supports, compute the corresponding projections, and find the sparse projection vector. TOO EXPENSIVE.
- If B is symmetric, then efficient computations methods exist.

The Permutation Group

Σ_n = permutation group of $[n]$

$\mathbf{x}^\sigma$ = reordering of $\mathbf{x}$ according to $\sigma \in \Sigma_n$,

$$(\mathbf{x}^\sigma)_i = x_{\sigma(i)}.$$

Example (permutation)

$\mathbf{x} = \begin{pmatrix} 5 & 4 & 6 \end{pmatrix}^T$, and

$$\sigma(1) = 3, \sigma(2) = 1, \sigma(3) = 2,$$

then

$$\mathbf{x}^\sigma = \begin{pmatrix} 6 & 5 & 4 \end{pmatrix}^T.$$

Symmetric Sets

- D is a **symmetric set** if

$$\mathbf{x} \in D \Rightarrow \mathbf{x}^\sigma \in D \ \ \forall \sigma \in \Sigma_n$$

set	description	sym.	nonneg. sym.	abs. sym.
Δ'_n [1]	unit sum	✓		
$[\ell, u]^n (\ell < u)$	box	✓		

[1] $\Delta'_n = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{1}^T \mathbf{x} = 1\}$

Nonnegative Symmetric Sets

- D is **nonnegative** if $\forall \mathbf{x} \in D$, $\mathbf{x} \geq \mathbf{0}$

set	description	sym.	nonneg. sym.	abs. sym.
$\mathbb{R}^n_+$	nonnegative orthant	$\checkmark$	$\checkmark$	
Δ_n	unit simplex	$\checkmark$	$\checkmark$	

Absolutely Symmetric Sets

- D is an **absolutely symmetric set** if it is symmetric and

$$\mathbf{x} \in D, \mathbf{y} \in \{-1, 1\}^n \Rightarrow \mathbf{x} \odot \mathbf{y} \equiv (x_i y_i)_{i=1}^n \in D$$

set	**description**	**sym.**	**nonneg. sym.**	**abs. sym.**
$\mathbb{R}^n$	entire space	✓		✓
$B_p[0,1] (p > 0)$	p-ball	✓		✓
C_s	s-sparse ball	✓		✓

Summary of Examples

set	desc.	sym.	non. sym.	abs. sym.
$\mathbb{R}^n$	entire space	✓		✓
$\mathbb{R}^n_+$	nonneg. orthant	✓	✓	
Δ_n	unit simplex	✓	✓	
$\Delta_n^{'}$	unit sum	✓		
$B_p[0,1] (p \geq 1)$	p-ball	✓		✓
C_s	s-sparse ball	✓		✓
$[\ell, u]^n (\ell < u)$	box	✓		

Symmetric Sparse Projection Theorem

Notation: given $\mathbf{x} \in \mathbb{R}^n$

$$\begin{aligned} M_k(\mathbf{x}) &= k \text{ indices corresponding to the } k \text{ largest values in } \mathbf{x} \\ L_k(\mathbf{x}) &= k \text{ indices corresponding to the } k \text{ smallest values in } \mathbf{x} \end{aligned}$$

Not uniquely defined.

Symmetric Sparse Projection Theorem B be a symmetric set, then a supper support of a vector $\exists \mathbf{y} \in P_{C_s \cap B}(\mathbf{x})$, $k \in \{0, \ldots, s\}$ for which

$$I_1(\mathbf{y}) \subseteq M_k(\mathbf{x}) \cup L_{s-k}(\mathbf{x})$$

Algorithm: Explore only $s+1$ supports.

Sparse Projection Onto Simple Symmetric Sets

- A set is called simple symmetric if it is either absolutely symmetric or nonnegative symmetric.

Sparse Projection Onto Simple Symmetric Sets

- A set is called simple symmetric if it is either absolutely symmetric or nonnegative symmetric.
- Given an underlying simple symmetric set, the symmetry function $p : \mathbb{R}^n \to \mathbb{R}^n$ is given by:

$$p_B(\mathbf{x}) = \begin{cases} \mathbf{x} & B \text{ is nonnegative symmetric,} \\ |\mathbf{x}| & B \text{ is absolutely symmetric.} \end{cases}$$

Sparse Projection Onto Simple Symmetric Sets

- A set is called simple symmetric if it is either absolutely symmetric or nonnegative symmetric.
- Given an underlying simple symmetric set, the symmetry function $p : \mathbb{R}^n \to \mathbb{R}^n$ is given by:

$$p_B(\mathbf{x}) \equiv \begin{cases} \mathbf{x} & B \text{ is nonnegative symmetric,} \\ |\mathbf{x}| & B \text{ is absolutely symmetric.} \end{cases}$$

Theorem (Sparse Projection onto Simple Symmetric Sets) Let B be a nonempty closed convex and simple symmetric set

Then

$$\exists \mathbf{y} \in P_{C_s \cap B}(\mathbf{x}) \text{ s.t. } I_1(\mathbf{y}) \subseteq M_S(p_B(\mathbf{x}))$$

Sparse Projection onto Simple Symmetric - Algorithm

Input: $\mathbf{x} \in \mathbb{R}^n$.

Output: $\mathbf{u} \in P_{B \cap C_s}(\mathbf{x})$.

1. Compute $T = M_s(p_B(\mathbf{x}))$.
2. Return $\mathbf{u}$: $\mathbf{u}_T = P_{B_T}(\mathbf{x}_T), \mathbf{u}_{T^c} = \mathbf{0}$.

Proximal Mapping of g_2

$g_2(\mathbf{x}) = \lambda\|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$
Sparse prox over B:

$$\mathrm{prox}_{g_2}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}} \left\{ \lambda\|\mathbf{y}\|_0 + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 : \mathbf{y} \in B \right\}$$

Proximal Mapping of g_2

$g_2(\mathbf{x}) = \lambda\|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$

Sparse prox over B:

$$\operatorname{prox}_{g_2}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}} \left\{ \lambda\|\mathbf{y}\|_0 + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 : \mathbf{y} \in B \right\}$$

- If $B = \mathbb{R}^n$, then $\operatorname{prox}_{g_2}(\mathbf{x})$ is the Hard Thresholding operator with level $\sqrt{2\lambda}$:

$$(\operatorname{prox}_{g_2}(\mathbf{x}))_i = \begin{cases} \{0\}, & |x_i| < \sqrt{2\lambda}, \\ \{x_i\}, & |x_i| > \sqrt{2\lambda}, \\ \{0, x_i\}, & |x_i| = \sqrt{2\lambda}. \end{cases}$$

Computing prox_{g_2} using prox_{g_1}

Underlying assumption: B is a simple symmetric set.

Computing prox_{g_2} using prox_{g_1}

Underlying assumption: B is a simple symmetric set.

- **Result:** a vector in prox_{g_2} can be evaluted by computing vectors in $P_{B \cap C_i}$ for any $i = 0, 1, \ldots, n$.

Computing prox_{g_2} using prox_{g_1}

Underlying assumption: B is a simple symmetric set.

- **Result:** a vector in prox_{g_2} can be evaluted by computing vectors in $P_{B\cap C_i}$ for any $i = 0, 1, \ldots, n$.
- **The projection sequence:**
  $P_B(\mathbf{x}; i) \in P_{B\cap C_i}(\mathbf{x}), \qquad T = M_i(p_B(\mathbf{x}))$

Underlying assumption: B is a simple symmetric set.

- **Result:** a vector in prox_{g_2} can be evaluted by computing vectors in $P_{B\cap C_i}$ for any $i = 0, 1, \ldots, n$.
- **The projection sequence:**
  $P_B(\mathbf{x}; i) \in P_{B\cap C_i}(\mathbf{x}), \qquad T = M_i(p_B(\mathbf{x}))$

Theorem. Any vector in

$$\mathrm{argmin}\left\{\lambda\|\mathbf{y}\|_0 + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 : \mathbf{y} \in \{P_B(\mathbf{x};0), ..., P_B(\mathbf{x};n)\}\right\}$$

is in $\mathrm{prox}_{g_2}(\mathbf{x})$

Computing prox_{g_2} using prox_{g_1}

Underlying assumption: B is a simple symmetric set.

- **Result:** a vector in prox_{g_2} can be evaluted by computing vectors in $P_{B\cap C_i}$ for any $i = 0, 1, \ldots, n$.
- **The projection sequence:**
  $P_B(\mathbf{x}; i) \in P_{B\cap C_i}(\mathbf{x}), \qquad T = M_i(p_B(\mathbf{x}))$

Theorem. Any vector in

$$\operatorname{argmin}\left\{\lambda\|\mathbf{y}\|_0 + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 : \mathbf{y} \in \{P_B(\mathbf{x};0), \ldots, P_B(\mathbf{x};n)\}\right\}$$

is in $\mathrm{prox}_{g_2}(\mathbf{x})$

Drawback: requires n projection computations.
Question: Can it be reduced to $O(\log n)$ computations?

Computing prox_{g_2} using prox_{g_1}

Underlying assumption: B is a simple symmetric set.

- **Result:** a vector in prox_{g_2} can be evaluted by computing vectors in $P_{B\cap C_i}$ for any $i = 0, 1, \ldots, n$.
- **The projection sequence:**
  $P_B(\mathbf{x}; i) \in P_{B\cap C_i}(\mathbf{x}), \qquad T = M_i(p_B(\mathbf{x}))$

Theorem. Any vector in

$$\mathrm{argmin}\left\{\lambda\|\mathbf{y}\|_0 + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 : \mathbf{y} \in \{P_B(\mathbf{x};0), ..., P_B(\mathbf{x};n)\}\right\}$$

is in $\mathrm{prox}_{g_2}(\mathbf{x})$

Drawback: requires n projection computations.
Question: Can it be reduced to $O(\log n)$ computations? Yes, under an additional assumption

The Second Order Monotonicity (SOM) Property

- **Definition.** A simple symmetric set $B \subseteq \mathbb{R}^n$ is said to satisfy the second order monotonicity property if $\forall \mathbf{x} \in \mathbb{R}^n, i \in \{0, 1, \ldots, n-2\}$it holds that

$$\|P_B(\mathbf{x}; i)-\mathbf{x}\|_2^2-\|P_B(\mathbf{x}; i+1)-\mathbf{x}\|_2^2 \geq \|P_B(\mathbf{x}; i+1)-\mathbf{x}\|_2^2-\|P_B(\mathbf{x}; i+2)-\mathbf{x}\|_2^2.$$

"The marginal gain in increasing the size of the support is decreasing"

The Second Order Monotonicity (SOM) Property

• **Definition.** A simple symmetric set $B \subseteq \mathbb{R}^n$ is said to satisfy the second order monotonicity property if $\forall \mathbf{x} \in \mathbb{R}^n, i \in \{0, 1, \ldots, n-2\}$it holds that

$$\|P_B(\mathbf{x}; i) - \mathbf{x}\|_2^2 - \|P_B(\mathbf{x}; i+1) - \mathbf{x}\|_2^2 \geq \|P_B(\mathbf{x}; i+1) - \mathbf{x}\|_2^2 - \|P_B(\mathbf{x}; i+2) - \mathbf{x}\|_2^2.$$

"The marginal gain in increasing the size of the support is decreasing"

• **Result 1.** Under the SOM property, a sparse prox vector can be found in $\lceil \log_2 n \rceil$ projections.

Sets Satisfying the SOM Property

Result 2. The following sets satisfy the SOM property:

Name of Set	Set
ℓ_∞-ball	$B_\infty[0, \alpha]$
nonnegative α-box	$[0, \alpha]^n$
–	$\mathbb{R}^n$
nonnegative orthant	$\mathbb{R}^n_+$
ℓ_2-ball	$B_2[0, \alpha]$
α-simplex	$\Delta_n(\alpha) = \{\mathbf{x} : \mathbf{e}^T\mathbf{x} = \alpha, \mathbf{x} \geq \mathbf{0}\}$
full α-simplex	$\Delta_n^F(\alpha) = \{\mathbf{x} : \mathbf{e}^T\mathbf{x} \leq \alpha, \mathbf{x} \geq \mathbf{0}\}$
ℓ_1-ball	$B_1[0, \alpha]$

Optimality Conditions and Algorithms

Back to the Sparse Optimization Problem

The sparse optimization model

$$(P) \quad \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

where either $g(\mathbf{x}) = g_1(\mathbf{x}) \equiv \delta_{B \cap C_s}(\mathbf{x})$ (model 1) or $g(\mathbf{x}) = g_2(\mathbf{x}) \equiv \lambda\|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$ (model 2)

Assumption

[A] *$f : \mathbb{R}^n \to \mathbb{R}$ is lower bounded, continuously differentiable.*

[B] *B is a simple symmetric closed and convex set.*

In some cases

[C] $f \in C^{1,1}_{L_f}$.

Road Map of Optimality Conditions

$$(P) \min f(\mathbf{x}) + g(\mathbf{x})$$

$g(\mathbf{x}) = g_1(\mathbf{x}) \equiv \delta_{B \cap C_s}(\mathbf{x})$ (model 1) or $g(\mathbf{x}) = g_2(\mathbf{x}) \equiv \lambda \|\mathbf{x}\|_0 + \delta_B(\mathbf{x})$ (model 2)

- Support Optimality - "optimality" over the support.
- L-Stationarity - extension of stationarity over convex sets.
- CW-optimality

To simplify the presentation - we will assume in the setting of model 1 ($g = g_1$) that all relevant points are with full support.

Support Optimality (SO)

- **Notation.** Set of optimal solutions over a given support $S \subseteq [n]$:

$$\mathcal{O}(S) = \operatorname*{argmin}_{\mathbf{u}}\{f(\mathbf{u}) : I_1(\mathbf{u}) \subseteq S, \mathbf{u} \in \mathrm{dom}(g)\}.$$

Support Optimality (SO)

- **Notation.** Set of optimal solutions over a given support $S \subseteq [n]$:

$$\mathcal{O}(S) = \operatorname*{argmin}_{\mathbf{u}} \{ f(\mathbf{u}) : I_1(\mathbf{u}) \subseteq S, \mathbf{u} \in \mathrm{dom}(g) \}.$$

A vector $\mathbf{x} \in \mathbb{R}^n$ is called support optimal if

$$\mathbf{x} \in \mathcal{O}(I_1(\mathbf{x})).$$

- **Theorem.** Any optimal solution is support optimal (no assumptions on B and f)

Support Optimality (SO)

- **Notation.** Set of optimal solutions over a given support $S \subseteq [n]$:

$$\mathcal{O}(S) = \operatorname*{argmin}_{\mathbf{u}}\{f(\mathbf{u}) : I_1(\mathbf{u}) \subseteq S, \mathbf{u} \in \mathrm{dom}(g)\}.$$

A vector $\mathbf{x} \in \mathbb{R}^n$ is called support optimal if

$$\mathbf{x} \in \mathcal{O}(I_1(\mathbf{x})).$$

- **Theorem.** Any optimal solution is support optimal (no assumptions on B and f)
- The condition can be verified if it is possible to minimize over restrictions of B (without the sparsity terms):

$$\min_{\mathbf{u}}\{f(\mathbf{u}) : \mathbf{u} \in B, u_i = 0, u \notin I_1(\mathbf{x})\}$$

Algorithm: How to find an SO point

- In model 1: Take $S \subseteq [n], |S| = s$ and compute $\mathbf{x} \in \mathcal{O}(S)$.
- In model 2: Take $S \subseteq [n]$ and compute $\mathbf{x} \in \mathcal{O}(S)$.

Algorithm: How to find an SO point

- In model 1: Take $S \subseteq [n], |S| = s$ and compute $\mathbf{x} \in \mathcal{O}(S)$.
- In model 2: Take $S \subseteq [n]$ and compute $\mathbf{x} \in \mathcal{O}(S)$.

- Exponential amount of SO points.
- Extremely weak condition.

Road Map of Optimality Conditions

$$(P) \quad \begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in C_s \cap B, \end{array}$$

- Support Optimality - "optimality" over the support.
- L-Stationarity - **extension of stationarity over convex sets.**
- CW-optimality

L-Stationarity

Unfortunately, the variational form $F'(\mathbf{x}^*, \mathbf{x} - \mathbf{x}^*) \geq 0 \forall \mathbf{x} \in \mathrm{dom}(g)$ is not a necessary optimality condition (in general...)

L-Stationarity

Unfortunately, the variational form $F'(\mathbf{x}^*, \mathbf{x} - \mathbf{x}^*) \geq 0 \forall \mathbf{x} \in \mathrm{dom}(g)$ is not a necessary optimality condition (in general...)

Let $L > 0$. A vector $\mathbf{x} \in \mathrm{dom}(g)$ is an ***L*-stationary point** of (P) if

$$\mathbf{x} \in \mathrm{prox}_{\frac{g}{L}} \left(\mathbf{x} - \frac{1}{L} \nabla f(\mathbf{x}) \right).$$

L-Stationarity

Unfortunately, the variational form $F'(\mathbf{x}^*, \mathbf{x} - \mathbf{x}^*) \geq 0 \forall \mathbf{x} \in \mathrm{dom}(g)$ is not a necessary optimality condition (in general...)

Let $L > 0$. A vector $\mathbf{x} \in \mathrm{dom}(g)$ is an ***L*-stationary point** of (P) if

$$\mathbf{x} \in \mathrm{prox}_{\frac{g}{L}}\left(\mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x})\right).$$

Example ($B = \mathbb{R}^n$)

$B = \mathbb{R}^n$, and $\sigma \in \tilde{\Sigma}(|\mathbf{x}^*|)$. Then $\mathbf{x}^*$ is an L-stationary point of (P) if and only if[a]

$$|\nabla_i f(\mathbf{x}^*)| \begin{cases} \leq L|x^*_{\langle s \rangle}| & \text{if } i \in I_0(\mathbf{x}^*), \\ = 0 & \text{if } i \in I_1(\mathbf{x}^*). \end{cases}$$

[a] Beck, A. & Eldar, Y. C., SIOPT, 2013

1. L-Stationarity $\Rightarrow$ SO (if f is convex)
2. If $f \in C^{1,1}_{L_f}$, Optimality $\Rightarrow$ L-stationarity $\forall L \geq L_f$

Condition depends on L, more restrictive as L gets smaller

Proximal Gradient Method

$$\mathbf{x}^{k+1} \in \operatorname{prox}_{\frac{g}{L}}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right)$$

Proximal Gradient Method

Proximal Gradient Method

$$\mathbf{x}^{k+1} \in \mathrm{prox}_{\frac{g}{L}}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right)$$

- $B = \mathbb{R}^n \Rightarrow$ Iterative Hard Thresholding (IHT) method (Blumensath and Davis '08, '09, '12).
- Makes sense only when $f \in C^{1,1}$.
- Only guarantees convergence to an L-stationary point for $L > L_f$.

Theorem. If $L > L_f$, then all limit points of the sequence generated by the PG method with stepsize $\frac{1}{L}$ are L-stationary points.

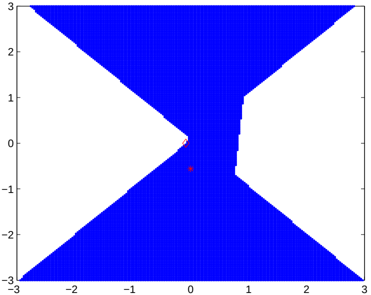
Back to L-stationarity - Example

$$\min\left\{f(x_1,x_2)\equiv 12x_1^2+20x_1x_2+32x_2^2 : \left\|(x_1;x_2)^T\right\|_0\leq 1\right\}$$

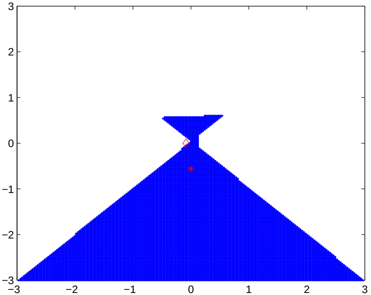
$$L_f = 48.3961$$

Two SO vectors: $(0,-9/16)$ - optimal solution. $(-1/12,0)$ - non-optimal, SL$=$196.

$L = 250$



$L = 500$



Road Map of Optimality Conditions

$$(P) \quad \begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in C_s \cap B, \end{array}$$

- Support Optimality - "optimality" over the support.
- L-Stationarity - extension of stationarity over convex sets.
- **CW-optimality**

Partial CW optimality

Lots of notions of "CW-optimality". We will concentrate on a "partial notion" where we compare the current point to (possibly) three points with similar support sets.

$$
\begin{aligned}
\mathbf{v}_{\mathbf{x}}^{-} &\in \mathcal{O}(I_1(\mathbf{x})\backslash\{i_{\mathbf{x}}\}),\\
\mathbf{v}_{\mathbf{x}}^{\text{swap}} &\in \mathcal{O}\left((I_1(\mathbf{x})\backslash\{i_{\mathbf{x}}\}) \cup \{j_{\mathbf{x}}\}\right),\\
\mathbf{v}_{\mathbf{x}}^{+} &\in \mathcal{O}(I_1(\mathbf{x}) \cup \{j_{\mathbf{x}}\})
\end{aligned}
$$

where

$$
\begin{aligned}
i_{\mathbf{x}} &\in \operatorname*{argmin}_{\ell \in C(\mathbf{x})}\{p_B(-\nabla_\ell f(\mathbf{x}))\} \text{ with } C(\mathbf{x}) = \operatorname*{argmin}_{k \in I_1(\mathbf{x})} p_B(x_k)\\
j_{\mathbf{x}} &\in \operatorname*{argmin}_{\ell \in I_0(\mathbf{x})}\left\{-p_B(-\nabla_\ell f(\mathbf{x}))\right\}.
\end{aligned}
$$

Partial CW-Optimality

- Model 1: (P_1) $\min\{F(\mathbf{x}) \equiv f(\mathbf{x}) : \mathbf{x} \in B \cap C_s\}$
- Model 2: (P_1) $\min\{F(\mathbf{x}) \equiv f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 : \mathbf{x} \in B\}$

Model 1: An SO point $\mathbf{x}^*$ is a coordinate-wise optimal point if

$$F(\mathbf{x}^*) \leq F(\mathbf{v}_{\mathbf{x}^*}^{\text{swap}})$$

Model 2: An SO point $\mathbf{x}^*$ is a coordinate-wise optimal point if

$$F(\mathbf{x}^*) \leq \min\{F(\mathbf{v}_{\mathbf{x}^*}^{\text{swap}}), F(\mathbf{v}_{\mathbf{x}^*}^{-}), F(\mathbf{v}_{\mathbf{x}^*}^{+})\}$$

Partial CW-Optimality in the Hierarchy

Results: I

1. Optimality $\Rightarrow$ partial CW-optimality
2. If $f \in C^{1,1}_{L_f}$, then partial CW-optimality $\Rightarrow$ L-stationarity $\forall L \geq L_f$

It can be shown that Partial CW-optimality actually implies L-stationarity for a smaller value than $L = L_f$

Partial CW-optimality is more restrictive than L_f-stationarity

Partial CW-Optimality in the Hierarchy

Results: I

1. Optimality $\Rightarrow$ partial CW-optimality
2. If $f \in C^{1,1}_{L_f}$, then partial CW-optimality $\Rightarrow$ L-stationarity $\forall L \geq L_f$

It can be shown that Partial CW-optimality actually implies L-stationarity for a smaller value than $L = L_f$

Partial CW-optimality is more restrictive than L_f-stationarity

A more restrictive condition: **full-CW optimality**. Loosely speaking, the point is better than any other point with a slightly different support set.

Hierarchy - Summary

Full CW-Optimality

$$\Downarrow$$

Partial CW-Optimality

$$\Downarrow$$

L_f-Stationarity

$$\Downarrow$$

Support Optimality

Numerical Example

$$\min_{\mathbf{x}\in\mathbb{R}^{10}} \{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + 0.2\|\mathbf{x}\|_0 : \|\mathbf{x}\|_1 \leq 1\}$$

supports	**support optimal**	***L*-stationary**	**partial CW**	**optimal**
1024	644	153	3	1

CD Method for Finding a CW-Optimal Point

Partial Coordinate Descent Method for Model 2:

1. **Initialization:** $\mathbf{x}^0 \in \mathbb{R}^n$ - an SO point. $k \leftarrow 0$;
2. set $\mathbf{x} = \mathbf{x}^k$ and compute $i_{\mathbf{x}}$ and $j_{\mathbf{x}}$.
3. compute

$$\begin{aligned} \mathbf{v}_{\mathbf{x}}^{-} &\in \mathcal{O}(I_1(\mathbf{x}) \backslash \{i_{\mathbf{x}}\}), \\ \mathbf{v}_{\mathbf{x}}^{-} &\in \mathcal{O}(I_1(\mathbf{x}) \cup \{j_{\mathbf{x}}\}), \\ \mathbf{v}_{\mathbf{x}}^{\mathrm{swap}} &\in \mathcal{O}\left((I_1(\mathbf{x}) \backslash \{i_{\mathbf{x}}\}) \cup \{j_{\mathbf{x}}\}\right). \end{aligned}$$

4. set $\mathbf{x}^{k+1} \in \operatorname{argmin}\left\{F(\mathbf{u}) : \mathbf{u} \in \{\mathbf{v}_{\mathbf{x}}^{-}, \mathbf{v}_{\mathbf{x}}^{\mathrm{swap}}, \mathbf{v}_{\mathbf{x}}^{+}\}\right\}$ (unless no improvement), $k \leftarrow k+1$, and go to step 2.

- Similar method exists for model 1.
- A full coordinate descent method can be defined that finds full CW-optimal points.

Hierarchy of Algorithms (Best to Worst)

- Full CD
- Partial CD
- Proximal Gradient

Numerical Example - Chances to Obtain the Optimum

$$\min \|\mathbf{Ax} - \mathbf{b}\|_2^2 + 0.5\|\mathbf{x}\|_0.$$

Monte Carlo Simulations (100 randomized initializations)

m	**n**	**s**	**PG**	**Partial CD**
32	320	2	13%	100%
64	640	2	5%	100%
96	960	2	42%	100%
128	1280	2	94%	100%
32	320	4	1%	70%
64	640	4	1%	99%
96	960	4	0%	100%
128	1280	4	0%	100%
32	320	6	0%	98%
64	640	6	0%	100%
128	1280	6	0%	100%
32	320	10	0%	0%
64	640	10	0%	90%
128	1280	10	0%	100%

THANK YOU FOR YOUR ATTENTION

- Beck, Hallak "On the Minimization Over Sparse Symmetric Sets: Projections, Optimality Conditions and Algorithms", *Mathematics of Operations Research*, vol. 41, no. 1 (2016) 196–223.
- Beck, Hallak "Proximal Mapping for Symmetric Penalty and Sparsity", *SIAM Journal on Optimization*, vol. 28, no. 1 (2018), 496–527.