CHARACTERIZATIONS OF DIFFERENTIABILITY, SMOOTHING TECHNIQUES AND DC PROGRAMMING WITH APPLICATIONS TO IMAGE RECONSTRUCTIONS

Nguyen Mau Nam[1], Daniel Giles[2], Le Thi Hoai An[3], Nguyen Thai An[4]

Abstract. In this paper, we study characterizations of differentiability for real-valued functions based on generalized differentiation. These characterizations provide the mathematical foundation for Nesterov's smoothing techniques in infinite dimensions. As an application, we provide a simple approach to image reconstructions based on Nesterov's smoothing techniques and DC programming that involves the $\ell_1 - \ell_2$ regularization.

Key words. DC programming; the DCA; Fenchel conjugate; denoising; inpainting.

AMS subject classifications. 49J52, 49J53, 90C31

1 Introduction and Problem Formulation

Gradient-based methods and second-order methods in optimization have been strongly developed over the last decades to solve optimization problems. One of the disadvantages of these methods is the requirement of the differentiability of the objective functions involved, while nondifferentiability appears frequently and naturally in many optimization models. A natural way to cope with the nondifferentiability in optimization is to approximate nonsmooth objective functions by smooth functions that are favorable for applying smooth optimization schemes. In his seminal paper [9], Nesterov proposed a method for approximating a class of nondifferentiable convex functions by smooth convex functions with Lipschitz continuous gradients. It turns out that this type of functions is highly important in solving nonsmooth optimization problems in many fields such as facility location, sparse optimization and compressed sensing.

The first goal of this paper is to further study Nesterov's smoothing techniques. Our idea comes from a well-known fact that convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ is Fréchet differentiable at $\bar{x} \in \mathbb{R}^n$ if and only if the subdifferential in the sense of convex analysis $\partial f(\bar{x})$ reduces to a singleton. We start by studying characterizations of differentiability for functions that are not necessarily convex in infinite dimensions. These characterizations allow us to study Nesterov's smoothing techniques in a more general setting, while having the potential of applying to broader classes of nondifferentiable functions usually considered in optimization.

[1]Fariborz Maseeh Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, USA (mau.nam.nguyen@pdx.edu). Research of this author was partly supported by the National Science Foundation under grant DMS-1716057.

[2]Department of Mathematics, Santa Barbara City College, Santa Barbara, CA, USA (djgiles1@pipeline.sbcc.edu).

[3]Computer Science and Applications Department, LGIPM, University of Lorraine, 57073 Metz, France (hoai-an.le-thi@univ-lorraine.fr).

[4]Institute of Research and Development, Duy Tan University, Vietnam (thaian2784@gmail.com). Research of this author was supported by the Vietnam National Foundation for Science and Technology Development under grant #101.01-2017.325.

Along with the difficulty in dealing with nondifferentiability, another challenge in modern optimization is to go from convexity to nonconvexity as nonconvex optimization techniques and algorithms allow us to solve more complex optimization problems arising naturally in many practical applications. This is a motivation of the search for new optimization methods to deal with broader classes of functions and sets where convexity is not assumed. One of the most successful approaches to go beyond convexity is to consider the class of DC functions, where DC stands for *difference of convex functions*. Given a linear space X, a DC program is an optimization problem in which we would like to minimize a function $f\colon X \to \mathbb{R}$ representable as $f = g - h$, where $g, h\colon X \to \mathbb{R}$ are convex functions. This extension of convex programming is not too far to take advantage of the available tools from convex analysis and optimization. At the same time, DC programming is sufficiently large to apply to many nonconvex optimization problems faced in recent applications.

Although the role of DC functions had been known earlier in optimization theory, the first algorithmic approach was developed by Pham Dinh Tao in 1985. The algorithm introduced by Pham Dinh Tao for minimizing $f = g - h$, called the DCA, is based on subgradients of the function h and subgradients of the Fenchel conjugate of the function g. This algorithm is summarized as follows: with given $x_1 \in \mathbb{R}^n$, define $y_k \in \partial h(x_k)$ and $x_{k+1} \in \partial g^*(y_k)$. Under suitable conditions on the DC decomposition of the function f, two sequences $\{x_k\}$ and $\{y_k\}$ in the DCA satisfy the monotonicity conditions in the sense that $\{g(x_k) - h(x_k)\}$ and $\{h^*(y_k) - g^*(y_k)\}$ are both decreasing. In addition, the sequences $\{x_k\}$ and $\{y_k\}$ converge to *critical points* of the primal function $g - h$ and the dual function $h^* - g^*$, respectively. The DCA usually depends on the choice of the starting point. However, with suitable initialization techniques, the DCA becomes very effective, producing sequences that converge to global solutions of the problem; see [10, 11] and the references therein.

In our recent research, we have been successful in applying Nesterov's smoothing techniques and the DCA to a number of optimization problems in facility location and clustering. This paper continues this effort by providing their applications to image reconstructions. Consider an unknown image M of size $N_1 \times N_2$. After the image is corrupted by a linear operator A and distorted by some noise ε, we observe only the image $b = A(M) + \varepsilon$, and seek to recover the true image M. The operator may act to simulate blurring, data compression, or down-sampling. In the case that the operator is the identity, the problem is called denoising.

We denote the columns of M as $m_1, \ldots, m_{N_2}$, to represent M in *vectorized form* as the $N_1N_2 \times 1$ column vector $M = [m_1^T\ m_2^T\ \cdots\ m_{N_2}^T]^T$. The vectorized form of M can be attained in MATLAB using the `reshape` function, and is equivalent to $\sum_{i=1}^{N_2}(e_i \otimes I_{N_1})(M \cdot e_i)$, where e_i is the i^{th} standard basis vector in $\mathbb{R}^{N_2}$ and $\otimes$ is the Kronecker product. Conversely, a vector $M \in \mathbb{R}^{N_1N_2}$ is reshaped into an $N_1 \times N_2$ matrix.

A vector is referred to as sparse when many of its entries are zero. An image $x \in \mathbb{R}^n$ (in vectorized form) is said to have a sparse representation y under D if there is some $n \times K$ matrix D, known as a dictionary, and a vector $y \in \mathbb{R}^K$ such that $x = Dy$. In this case the dictionary D maps a sparse vector to a full image. The columns of D are called *atoms*, and given a suitable dictionary in this model, theoretically any image can be built from a linear combination of the columns (atoms) of the dictionary. Using a clever choice of dictionary allows us to work with sparse vectors, thereby reducing the amount of computer memory needed to store an image. Further, sparse representations tend to capture the true image without extraneous noise.

The method in this paper is based on the following foundational model: Given a dictionary D and an observed image b which has been corrupted by a linear operator A, recover a sparse representation of the image by solving the minimization problem

$$\min_{y} \|y\|_0 \quad \text{s.t.} \quad \|A(Dy) - b\|^2 \leq \varepsilon$$

where $\varepsilon > 0$ is some small constraint term. Here $\|y\|_0$ is not a norm, but simply the number of nonzero entries in y. It is common to approximate $\|\cdot\|_0$ with $\|\cdot\|_1$, or with $\|\cdot\|_1 - \|\cdot\|$, known as ℓ_1 and $(\ell_1 - \ell_2)$ regularization, respectively. In this paper, using the DCA and Nesterov's smoothing techniques, we develop a very simple algorithms based on the $(\ell_1 - \ell_2)$ regularization for image reconstructions. The proposed method allows us to avoid solving subproblems in using the DCA for the $(\ell_1 - \ell_2)$ regularization; see [12, 13] and the references therein. We also apply this idea to build a simple but effective algorithm for dictionary learning. Our numerical examples show that our algorithms are competitive with state-of-the-art methods for image reconstructions.

2 Characterizations of Differentiability and Nesterov's Smoothing Techniques

In this section, we study characterizations of strict differentiability and their applications to smoothing techniques. Consider a real normed space X with its topological dual denoted by X^* which consists of all real-valued linear continuous functions defined on X. It is well-known that X^* is a normed space with the dual norm given by

$$\|x^*\| = \sup\{\langle x^*, x\rangle \mid \|x\| \leq 1\}, x^* \in X^*,$$

where $\langle x^*, x\rangle = x^*(x)$.

Let $f\colon X \to \overline{\mathbb{R}} = (-\infty, \infty]$ be a extended-real-valued function with the effective domain $\operatorname{dom}(f) = \{x \in X \mid f(x) < \infty\}$ and $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$. We first recall some classical concepts of differentiability. We say that f is *Gâteaux differentiable* at $\bar{x}$ if there exists $x^* \in X^*$ such that

$$\lim_{t \to 0+} \frac{f(\bar{x} + td) - f(\bar{x}) - t\langle x^*, d\rangle}{t} = 0 \text{ for all } d \in X.$$

Such an element x^* is unique if exists and is called the *Gâteaux derivative* of f at $\bar{x}$ denoted by $\nabla_G f(\bar{x})$. It follows directly from the definition that f is Gâteaux differentiable at $\bar{x}$ with $\nabla_G f(\bar{x}) = x^*$ if and only if

$$\lim_{t \to 0} \frac{f(\bar{x} + td) - f(\bar{x}) - t\langle x^*, d\rangle}{t} = 0 \text{ for all } d \in X.$$

We say that f is *Fréchet differentiable* at $\bar{x}$ if there exists $x^* \in X^*$ such that

$$\lim_{h \to 0} \frac{f(\bar{x}+h) - f(\bar{x}) - \langle x^*, h\rangle}{\|h\|} = 0.$$

The element x^* is called the *Fréchet derivative* of f at $\bar{x}$ denoted by $\nabla_F f(\bar{x})$. It follows from the definition that if f is Fréchet differentiable at $\bar{x}$, then it is Gâteaux differentiable at this point.

Let us now discuss a stronger concept of differentiability compared with the Gâteaux differentiability. We say that f is *Hadamard strictly differentiable* if there exists $x^* \in X^*$ such that for each $d \in X$,

$$\lim_{x \to \bar{x}, t \to 0^+} \frac{f(x+td) - f(x) - t\langle x^*, d\rangle}{t} = 0,$$

where the convergence is uniform for d in compact subsets of X. The last condition is automatic in the case where f is locally Lipschitz continuous around $\bar{x}$ as in the proposition below; see [3, Proposition 2.2.1].

Proposition 2.1 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ with $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$ and $x^* \in X^*$. Then the following properties are equivalent:*

(a) *f is Hadamard strictly differentiable at $\bar{x}$ and $\nabla_G f(\bar{x}) = x^*$.*

(b) *f is locally Lipschitz continuous around $\bar{x}$, and for each $d \in X$ one has*

$$\lim_{x \to \bar{x}, t \to 0^+} \frac{f(x+td) - f(x) - t\langle x^*, d\rangle}{t} = 0.$$

We also say that f is *Fréchet strictly differentiable* at $\bar{x}$ if there exists $x^* \in X^*$ such that

$$\lim_{x, y \to \bar{x}, x \neq y} \frac{f(x) - f(y) - \langle x^*, x - y\rangle}{\|x - y\|} = 0.$$

Note that if f is Fréchet strictly differentiable at $\bar{x}$, then it is both Fréchet differentiable and Hadamard strictly differentiable at this point.

Given a function $f\colon X \to \overline{\mathbb{R}}$ that is locally Lipschitz continuous around $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$, the *Clarke generalized directional derivative* of f at $\bar{x}$ in the direction $d \in X$ is defined by

$$f^\circ(\bar{x}; d) = \limsup_{x \to \bar{x}, t \to 0^+} \frac{f(x+td) - f(x)}{t}.$$

Based on the generalized derivative of f at $\bar{x}$, the *Clarke subdifferential* of f at $\bar{x}$ is defined by

$$\partial_c f(\bar{x}) = \{x^* \in X^* \mid \langle x^*, d\rangle \le f^\circ(\bar{x}; d) \text{ for all } d \in X\}.$$

Note that in the case where f is convex,

$$\partial_c f(\bar{x}) = \partial f(\bar{x}) = \{x^* \in X^* \mid \langle x^*, x - \bar{x}\rangle \le f(x) - f(\bar{x}) \text{ for all } x \in X\},$$

which is the *subdifferential in the sense of convex analysis* of f at $\bar{x}$. A characterization for Hadamard strict differentiabily is given in the proposition below; see [3, Proposition 2.2.4].

Proposition 2.2 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ with $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$. If f is Hadamard strictly differentiable at $\bar{x}$, then f is locally Lipschitz continuous around $\bar{x}$ and $\partial_c f(\bar{x}) = \{\nabla_G f(\bar{x})\}$. Conversely, if f is locally Lipschitz continuous around $\bar{x}$ and $\partial_c f(\bar{x})$ reduces to a singleton $\{x^*\}$, then f is Hadamard strictly differentiable at $\bar{x}$ and $\nabla_G f(\bar{x}) = x^*$.*

The following is a direct consequence of this result for the convex case.

Corollary 2.3 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ be a convex function with $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$. Then the following properties are equivalent:*

(a) *f is Hadamard strictly differentiable at $\bar{x}$.*

(b) *f is locally Lipschitz continuous around $\bar{x}$ and f is Gâteaux differentiable $\bar{x}$.*

(c) *f is locally Lipschitz continuous around $\bar{x}$ and $\partial f(\bar{x})$ is a singleton.*

In what follows we study a characterization for Fréchet differentiability based on Clarke subdifferentials. For a nonempty subset $\emptyset$ and $\bar{x} \in X$, the notation $d(\bar{x}; \emptyset)$ is used for the distance from $\bar{x}$ to $\emptyset$ defined by

$$d(\bar{x}; \emptyset) = \inf\{\|\bar{x} - w\| \mid w \in \emptyset\}.$$

Given a set-valued mapping $F\colon X \rightrightarrows X^*$, where both X and X^* are equipped with the strong topology. We say that F is *upper semicontinuous* at $\bar{x} \in \operatorname{dom}(F) := \{x \in X \mid F(x) \neq \emptyset\}$ if for any $\varepsilon > 0$, there exists $\delta > 0$ such that

$$F(x) \subset \mathbb{B}(F(\bar{x}); \varepsilon) \text{ whenever } x \in \mathbb{B}(\bar{x}; \delta),$$

where $\mathbb{B}(F(\bar{x}); \varepsilon) = \{x^* \in X^* \mid d(x^*; F(\bar{x})) \leq \varepsilon\}$ and $\mathbb{B}(\bar{x}; \delta)$ denotes the closed ball with center $\bar{x}$ and radius δ.

The following version of the mean value theorem (see [3]) is useful in what follows.

Theorem 2.4 *Let X be a normed space and let $f\colon X \to \mathbb{R}$ be Lipschitz continuous on an open set $G \subset X$. For any $[a, b] \subset G$, there exists $z \in (a, b)$ such that*

$$f(b) - f(a) \in \langle \partial_c f(z), b - a \rangle.$$

Let us now present a characterization of Fréchet strict differentiability based on Clarke subdifferentials; see, e.g., [4]. We provide an alternative detailed proof here for the convenience of the reader.

Theorem 2.5 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ with $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$. Then the following properties are equivalent:*

(a) *f is Fréchet strictly differentiable at $\bar{x}$.*

(b) *f is locally Lipschitz continuous around $\bar{x}$, $\partial_c f(\bar{x})$ is a singleton, and $\partial_c f(\cdot)$ is upper semicontinuous at $\bar{x}$.*

Proof. (a) $\Longrightarrow$ (b): Suppose that f is Fréchet strictly differentiable at $\bar{x}$. It is not hard to show that f is locally Lipschitz continuous around $\bar{x}$. Let us first show that $\partial_c f(\bar{x})$ is a singleton. Indeed,

$$f^\circ(\bar{x}; v) = \limsup_{x \to \bar{x}, t \to 0^+} \frac{f(x + tv) - f(x)}{t} = \langle \nabla_F f(\bar{x}), v \rangle \text{ for all } v \in X.$$

Fix any $x^* \in \partial_c f(\bar{x})$. Then $\langle x^*, v \rangle \leq \langle \nabla_F f(\bar{x}), v \rangle$ for all $v \in X$. This implies that $x^* = \nabla_F f(\bar{x})$. It remains to show that $\partial_c f(\cdot)$ is upper semicontinuous at $\bar{x}$. Fix any sequence $\{x_k\}$ in X that converges to $\bar{x}$, and fix any $x_k^* \in \partial_c f(x_k)$. Fix any $\varepsilon > 0$. Then there exists $\delta > 0$ such that

$$|f(x) - f(u) - \langle \nabla_F f(\bar{x}), x - u \rangle| \leq \varepsilon \|x - u\| \text{ whenever } x, u \in \mathbb{B}(\bar{x}; \delta).$$

Since $\{x_k\}$ converges to $\bar{x}$, we can find $k_0 \in \mathbb{N}$ such that $x_k \in \mathbb{B}(\bar{x}; \delta/4)$ for all $k \geq k_0$. Fix any $k \geq k_0$ and any $v \in X$. If $\|x - x_k\| < \delta/4$ and $0 < t < \delta/(2(\|v\| + 1))$, then $\|x - \bar{x}\| \leq \|x - x_k\| + \|x_k - \bar{x}\| < \delta/4 + \delta/4 = \delta/2 < \delta$. It follows that $\|x + tv - \bar{x}\| \leq \|x - \bar{x}\| + t\|v\| < \delta/2 + \delta/2 = \delta$, and so

$$f(x + tv) - f(x) \leq \langle \nabla_F f(\bar{x}), tv \rangle + \varepsilon \|tv\|.$$

This implies

$$\frac{f(x + tv) - f(x)}{t} \leq \langle \nabla_F f(\bar{x}), v \rangle + \varepsilon \|v\| \text{ for such } x, t.$$

It follows that

$$f^\circ(x_k; v) = \limsup_{t \to 0+, x \to x_k} \frac{f(x + tv) - f(x)}{t} \leq \langle \nabla_F f(\bar{x}), v \rangle + \varepsilon \|v\|.$$

Now, $\langle x_k^*, v \rangle \leq f^\circ(x_k; v) \leq \langle \nabla_F f(\bar{x}), v \rangle + \varepsilon \|v\|$. Then $\|x_k^* - \nabla_F f(\bar{x})\| \leq \varepsilon$ for all $k \geq k_0$. Therefore, $\{x_k^*\}$ converges strongly to $\nabla_F f(\bar{x})$. The upper semicontinuity of the Clarke subdifferential mapping is now straightforward.

(b) $\Longrightarrow$ (a): Let us prove the converse by assuming that (b) is satisfied. Suppose that $\partial_c f(\bar{x}) = \{x^*\}$. Since $\partial_c f(\cdot)$ is upper semicontinuous at $\bar{x}$, for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$\partial_c f(u) \subset \mathbb{B}(x^*, \varepsilon) \text{ whenever } u \in \mathbb{B}(\bar{x}, \delta).$$

We can choose $\delta > 0$ sufficiently small such that f is Lipschitz continuous on the open ball $B(\bar{x}, \delta)$. Fix any $x, y \in B(\bar{x}; \delta)$ with $x \neq y$. By the subdifferential mean value theorem, there exist $u \in (x, y)$ and $w^* \in \partial_c f(u)$ such that

$$f(x) - f(y) = \langle w^*, x - y \rangle.$$

Then $\|w^* - x^*\| \leq \varepsilon$, and hence

$$\left| \frac{f(x) - f(y) - \langle x^*, x - y \rangle}{\|x - y\|} \right| = \left| \frac{\langle w^* - x^*, x - y \rangle}{\|x - y\|} \right| \leq \|w^* - x^*\| \leq \varepsilon.$$

Therefore, f is Fréchet strictly differentiable at $\bar{x}$. $\square$

Let us now derive a corollary for the convex case.

Corollary 2.6 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ be a convex function with $\bar{x} \in \operatorname{int}(\operatorname{dom}(f))$. Then the following properties are equivalent:*

(a) *f is Fréchet strictly differentiable at $\bar{x}$.*

(b) *f is Fréchet differentiable at $\bar{x}$.*

(c) *f is locally Lipschitz continuous around $\bar{x}$, $\partial f(\bar{x})$ is a singleton, and $\partial f(\cdot)\colon X \rightrightarrows X^*$ is upper semicontinuous at $\bar{x}$.*

Proof. The implication (a) $\Longrightarrow$ (b) is obvious. If f is Fréchet differentiable at $\bar{x}$, it is well-known that $\partial_c f(\bar{x}) = \partial f(\bar{x}) = \{\nabla_F f(\bar{x})\}$ under the convexity of f. In addition, f is locally bounded around $\bar{x}$, so it is locally Lipschitz continuous around this point. Let us show that the subdifferential mapping is upper semicontinuous at $\bar{x}$. Let $x^* = \nabla_F f(\bar{x})$. We have

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \langle x^*, x - \bar{x}\rangle}{\|x - \bar{x}\|} = 0.$$

For any $\varepsilon > 0$, we can choose $\delta > 0$ such that

$$\langle x^*, x - \bar{x}\rangle \le f(x) - f(\bar{x}) + \varepsilon\|x - \bar{x}\| \text{ whenever } \|x - \bar{x}\| < \delta.$$

Fix any $x \in \mathbb{B}(\bar{x}; \delta)$ and any $u^* \in \partial f(x)$. Then

$$\langle x^* - u^*, x - \bar{x}\rangle = \langle x^*, x - \bar{x}\rangle + \langle u^*, \bar{x} - x\rangle \le f(x) - f(\bar{x}) + \varepsilon\|x - \bar{x}\| + f(\bar{x}) - f(x) = \varepsilon\|x - \bar{x}\|.$$

This implies $\|x^* - u^*\| \le \varepsilon$, which justifies the upper semicontinuity of the subdifferential mapping. Thus, the implication (b)$\Longrightarrow$ (c) holds. Finally, the implication (c)$\Longrightarrow$ (a) follows from Theorem 2.5. $\square$

Given a function $f\colon X \to \overline{\mathbb{R}}$, recall that the *Fenchel conjugate* of f is given by

$$f^*(x^*) = \sup\{\langle x^*, x\rangle - f(x) \mid x \in X\}, x^* \in X^*.$$

Proposition 2.7 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ be proper (i.e.,* $\operatorname{dom}(f) \neq \emptyset$*) and strongly coercive in the sense that*

$$\frac{\gamma}{2}\|x\|^2 + \langle u^*, x\rangle + b \le f(x) \text{ for all } x \in X, \tag{2.1}$$

for some $u^* \in X^*$, $b \in \mathbb{R}$, and $\gamma > 0$. Then $\operatorname{dom}(f^*) = X^*$ *and f^* is continuous on X^*, where X^* is equipped with the strong topology.*

Proof. Fix any $x^* \in X^*$. Then

$$-\infty < f^*(x^*) = \sup\{\langle x^*, x\rangle - f(x) \mid x \in X\} \le \sup\{-\frac{\gamma}{2}\|x\|^2 + \langle x^* - u^*, x\rangle - b \mid x \in X\} < \infty.$$

This implies that $\operatorname{dom}(f^*) = X^*$. Since $f^*\colon X \to \mathbb{R}$ is a convex function, to prove the continuity of f on X^*, it suffices to prove that f is locally bounded above on a neighborhood of the origin of X^* with respect to the strong topology. Fix any $\delta > 0$ and $v^* \in X^*$ with $\|v^*\| \leq \delta$. Then

$$\begin{aligned} f^*(v^*) &\leq \sup\{-\frac{\gamma}{2}\|x\|^2 + \langle v^* - u^*, x\rangle - b \mid x \in X\} \\ &\leq \sup\{-\frac{\gamma}{2}\|x\|^2 + \|v^* - u^*\|\|x\| - b \mid x \in X\} \\ &\leq \sup\{-\frac{\gamma}{2}\|x\|^2 + (\|u^*\| + \delta)\|x\| - b \mid x \in X\} < \infty. \end{aligned}$$

It follows that f is continuous at the origin, and so it is continuous on $\operatorname{int}(\operatorname{dom}(f)) = X^*$. □

Definition 2.8 *Let X be a normed space and let $f\colon X \to \overline{\mathbb{R}}$ be a proper convex function. We say that $\partial f(\cdot)$ is strongly monotone with parameter $\sigma > 0$ if*

$$\sigma\|x_1 - x_2\|^2 \leq \langle v_1^* - v_2^*, x_1 - x_2\rangle \text{ whenever } v_i^* \in \partial\varphi(x_i), i = 1, 2.$$

In particular,

$$\sigma\|x_1 - x_2\| \leq \|v_1^* - v_2^*\| \text{ whenever } v_i^* \in \partial\varphi(x_i), i = 1, 2.$$

Theorem 2.9 *Let X be a reflexive Banach space and let $f\colon X \to \overline{\mathbb{R}}$ be a proper lower semicontinuous function. Suppose that f is strictly convex and strongly coercive. Then f^* is Gâteaux differentiable. If we assume in addition that $\partial f(\cdot)$ is strongly monotone with parameter $\sigma > 0$, then f^* is Fréchet differentiable and $\nabla_F f^*$ is Lipschitz continuous with constant $\ell = 1/\sigma$.*

Proof. By Proposition 2.7, the function f^* is convex and continuous on X^*. Fix any $v^* \in X^*$. Let us first prove the Gâteaux differentiability of f^* at v^* by using Corollary 2.3. Note that $\bar{x} \in \partial f^*(v^*)$ if and only if $v^* \in \partial f(\bar{x})$, which holds iff

$$f(\bar{x}) - \langle v^*, \bar{x}\rangle \leq f(x) - \langle v^*, x\rangle \text{ for all } x \in X.$$

Equivalently, $\bar{x}$ is an absolute minimizer of the function $g(x) = f(x) - \langle v^*, x\rangle$ for $x \in X$. It follows from (2.1) that $\lim_{\|x\|\to\infty} g(x) = \infty$. Then by the strict convexity of f, the function g has a unique absolute minimizer on X. Thus, $\partial f^*(v^*) = \{\bar{x}\}$ is a singleton. Therefore, by Corollary 2.3, the function f is Gâteaux differentiable at v^*.

Now, we assume in addition that that $\partial f(\cdot)$ is strongly monotone with parameter $\sigma > 0$. Fix any $v_i^* \in X^*$ and $x_i \in X$ with $x_i \in \partial f^*(v_i^*)$ for $i = 1, 2$. Then $v_i^* \in \partial f(x_i)$ for $i = 1, 2$ and

$$\|x_1 - x_2\| \leq \frac{1}{\sigma}\|v_1^* - v_2^*\|.$$

Thus, we can easily show that the subdifferential mapping $\partial f(\cdot)\colon X \rightrightarrows X^*$ is upper semicontinuous at v^*. It follows from Theorem 2.6 that f^* is Fréchet differentiable, and in addition,

$$\|\nabla_F f(v_1^*) - \nabla_F f(v_2^*)\| \leq \frac{1}{\sigma}\|v_1^* - v_2^*\|.$$

for all $v_1^*, v_2^* \in X^*$. This completes the proof. $\square$

For a bounded linear mapping $A\colon X \to Y$ between normed spaces, we define the norm of A as usual:

$$\|A\| = \sup\left\{\|A(x)\| \;\middle|\; \|x\| \le 1\right\}.$$

It follows from the definition that $\|A(x)\| \le \|A\|\|x\|$ for all $x \in X$. The adjoint mapping of A denoted by $A^*\colon Y^* \to X^*$ is defined by $A^*(y^*) = y^* \circ A$ for $y^* \in Y^*$. It is well-known that if $A\colon X \to Y$ is a bounded linear mapping, then $\|A\| = \|A^*\|$.

Lemma 2.10 *Let $A\colon X \to Y^*$ be a bounded linear mapping, where Y is a reflexive Banach space, let $\varphi\colon Y \to \mathbb{R}$ be a continuous function, and let Q be a nonempty closed convex set in Y. If φ is strictly convex and strongly coercive, then the function $g\colon X \to \mathbb{R}$ defined by*

$$g(x) = \sup\{\langle Ax, y\rangle - \varphi(y) \mid y \in Q\}, x \in X,$$

is well-defined and Gâteaux differentiable. If we assume in addition that $\partial\varphi(\cdot)$ is strongly monotone with parameter $\sigma > 0$, then g is Fréchet differentiable. In addition, $\nabla_F g$ is Lipschitz continuous with constant $\ell = \|A\|^2/\sigma$.

Proof. The function g can be represented as

$$g(x) = \sup\{\langle Ax, y\rangle - [\varphi(y) + \delta_Q(y)] \mid y \in Y\} = \sup\{\langle Ax, y\rangle - h(y) \mid y \in Y\} = h^*(Ax),$$

where the function $h\colon Y \to \overline{\mathbb{R}}$ defined by $h(y) = \varphi(y) + \delta_Q(y)$ for $y \in Y$. Based on the strong coercivity of h, we can easily show that

$$\lim_{\|y\|\to\infty} (\langle Ax, y\rangle - h(y)) = -\infty.$$

Thus, the function g is well-defined and the “supremum” in its definition becomes “max”.

If φ is strictly convex and strongly coercive, then so is h. By Theorem 2.9, the function h^* is Gâteaux differentiable. Thus, it is straightforward to see that g is Gâteaux differentiable. Now, assume that $\partial\varphi(\cdot)$ is strongly monotone with parameter σ. By the subdifferential sum rule, we can show that $\partial h(\cdot)$ is also strongly monotone with parameter σ. It follows from Theorem 2.9 that the function h^* is Fréchet differentiable and $\nabla_F h^*$ is Lipschitz continuous with constant $1/\sigma$. Then g is Fréchet differentiable and $\nabla_F g$ is Lipschitz continuous with constant $\ell = \|A\|^2/\sigma$. $\square$

Let X and Y be normed spaces. Given a bounded linear mapping $A\colon X \to Y^*$, a continuous convex function $\phi\colon Y \to \mathbb{R}$, and a nonempty closed convex set $Q \subset Y$, consider the function

$$f(x) = \sup\{\langle Ax, y\rangle - \phi(y) \mid y \in Q\}, x \in X. \tag{2.2}$$

In general, $f\colon X \to \overline{\mathbb{R}}$ is a nondifferentiable convex function.

Our goal now is to find a differentiable approximation of the function f given by (2.2). The idea comes from [9] as follows. Fix a continuous function $d\colon Y \to [0, \infty)$. Given $\mu > 0$, define

$$f_\mu(x) = \sup\{\langle Ax, y\rangle - \phi(y) - \mu d(y) \mid y \in Q\}, x \in X. \tag{2.3}$$

Theorem 2.11 *Let X be a normed space and let Y be a reflexive Banach space. Consider the function f defined in (2.2) and the function f_μ defined in (2.3).*

(a) *Suppose that d is strictly convex and strongly coercive, then f_μ is Gâteaux differentiable.*

(b) *If we assume further that $\partial d(\cdot)$ is strongly monotone with parameter $\sigma > 0$, then f_μ is a $C^{1,1}$ function. In addition, ∇f_μ is Lipschitz continuous with constant*

$$\ell_\mu = \frac{\|A\|^2}{\sigma\mu}.$$

We also have the estimate

$$f_\mu(x) \le f(x) \le f(x) + \mu \sup_{y\in Y} d(y) \text{ for all } x \in X.$$

Proof. The conclusion follows directly from Lemma 2.10. Note that if d is strictly convex and strongly coercive, then the function $\varphi(u) = \phi(u) + \mu d(u)$ for $u \in Y$ is also strictly convex and strongly coercive. If, in addition, $\partial d(\cdot)$ is strongly monotone with parameter $\sigma > 0$, then $\partial\varphi(\cdot)$ is strongly monotone with parameter $\sigma\mu$. For any $x \in X$,

$$f_\mu(x) = \sup\{\langle Ax, y\rangle - \phi(y) - \mu d(y) \mid y \in Q\} \le \sup\{\langle Ax, y\rangle - \phi(y) \mid y \in Q\} = f(x).$$

We also have

$$\begin{aligned} f(x) &= \sup\{\langle Ax, y\rangle - \phi(y) \mid y \in Q\} \\ &= \sup\{\langle Ax, y\rangle - \phi(y) - \mu d(y) + \mu d(y) \mid y \in Q\} \\ &\le \sup\{\langle Ax, y\rangle - \phi(y) - \mu d(y) \mid y \in Q\} + \sup_{y\in Q}(\mu d(y)) \le f_\mu(x) + \mu \sup_{y\in Q} d(u). \end{aligned}$$

The proof is now complete. □

Let us continue by providing some examples of the function d that satisfies condition (a) or (b) in Theorem 2.11. Recall that a subset F with nonempty interior of a normed space is called *strictly convex* if for any $x, y \in F$ with $x \neq y$ and for any $t \in (0,1)$, we have $tx + (1-t)y \in \text{int}(F)$. The proof of the proposition below is straightforward.

Proposition 2.12 *Let X be a normed space amd let F be a nonempty convex set in X that contains the origin in its interior. Suppose that F is strictly convex. Consider the Minkowski function associated with F defined by $\rho_F(x) = \inf\{t > 0 \mid x \in tF\}$ for $x \in X$. Then the function $d = (\rho_F)^2$ is continuous, strictly convex, and strongly coercive.*

We say that a function $d\colon X \to \overline{\mathbb{R}}$ is called $F-$*strongly convex* with parameter $\sigma > 0$ if the function $d - \sigma/2(\rho_F)^2$ is convex. In particular, if d is $\mathbb{B}-$strongly convex, where $\mathbb{B}$ is the closed unit ball of X, then this definition reduces to the well-known definition of strong convexity.

Proposition 2.13 *Let X be a normed space and let $d\colon X \to \mathbb{R}$ be a continuous function that is $F-$strongly convex, where the set F satisfies the conditions in Proposition 2.12.*

Then d is also strictly convex and strongly coercive. If we assume in addition that X is a Hilbert space and d is strongly convex with parameter $\sigma > 0$, then $\partial d(\cdot)$ is strongly monotone with parameter σ.

Proof. Define the function $h = d - \sigma/2(\rho_F)^2$. Then h is a continuous convex function. Thus, there exist $w^* \in X^*$ and $b \in \mathbb{R}$ such that

$$\langle w^*, x\rangle + b \leq h(x) \text{ for all } x \in X.$$

Since $d = h + \sigma/2(\rho_F)^2$, the conclusions become straightforward.

Finally, let us consider a direct corollary of Theorem 2.11 (b).

Corollary 2.14 *Let X and Y be Hilbert spaces. In the setting of Theorem 2.11, let $d(y) = 1/2\|y - y_0\|^2$ for $y \in Y$, where $y_0 \in Q$ and Q is bounded. The function f_μ given by (2.3) is Fréchet differentiable and its gradient is Lipschitz continuous on X with Lipschitz constant $\ell = \|A\|^2/\mu$. In addition,*

$$f_\mu(x) \leq f_0(x) \leq f_\mu(x) + \frac{\mu}{2}[D(y_0; Q)]^2 \text{ for all } x \in X,$$

where $D(y_0; Q) = \sup\{\|y_0 - y\| \mid y \in Q\} < \infty$.

In particular, if $\psi(y) = \langle b, y\rangle$ for $y \in Y$, where $b \in Y$, then the function f_μ has the explicit representation

$$f_\mu(x) = \frac{\|Ax - b\|^2}{2\mu} + \langle Ax - b, y_0\rangle - \frac{\mu}{2}\big[d(y_0 + \frac{Ax - b}{\mu}; Q)\big]^2$$

and is Fréchet differentiable on X with its gradient given by

$$\nabla f_\mu(x) = A^* u_\mu(x),$$

where u_μ can be expressed in terms of the Euclidean projection

$$u_\mu(x) = \pi(y_0 + \frac{Ax - b}{\mu}; Q).$$

The gradient ∇f_μ is a Lipschitz function with constant

$$\ell_\mu = \frac{1}{\mu}\|A\|^2.$$

Proof. The conclusion follows directly from Theorem 2.11 with the observation that d is strongly convex with constant $\sigma = 1$; see [8] for more details. □

3 Applications to Image Reconstructions

In this section, we consider an unknown image M of size $N_1 \times N_2$. After the image is corrupted by a linear operator A and distorted by some noise ε, we observe only the image $b = A(M) + \varepsilon$, and seek to recover the true image M.

3.1 Patching an Image

In this section, we describe an optimization problem which models the image reconstruction problem by expressing an image as the sum of sparse representations of distinct 'patches' of the image; see, e.g., [15]. Given an $N_1 \times N_2$ image matrix M, let $\mathcal{P}$ be a collection of submatrices $P_{i,j}$ of M with size $n_1 \times n_2$, which cover M. We henceforth refer to these submatrices as patches, and sometimes identify $P_{i,j}$ with its index (i,j). The covering condition ensures that every pixel of M appears in some patch, and we will use collections only of non-overlapping patches, so that $\mathcal{P}$ partitions M.

$$R_{1,1} \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline 6 & 7 & 8 & 9 & 10 \\ \hline 11 & 12 & 13 & 14 & 15 \\ \hline 16 & 17 & 18 & 19 & 20 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 6 & 7 \\ \hline \end{array} \qquad R_{1,1}^T \begin{array}{|c|c|} \hline w & x \\ \hline y & z \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline w & x & 0 & 0 & 0 \\ \hline y & z & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

Figure 1: *Left*: $R_{1,1}$ acts on a 4×5 matrix to extract the 4×4 patch $(1,1)$. *Right*: $R_{1,1}^T$ embeds a 4×4 patch into patch $(1,1)$ of a zero matrix.

Define $R_{i,j}$ as the function that maps the image M to patch $P_{i,j}$, that is, $R_{i,j}(M) = P_{i,j}$. If M is in vectorized form, $R_{i,j}$ can be expressed as an $n_1 n_2 \times N_1 N_2$ matrix with exactly one 1 in each row and zeros elsewhere. In particular, $[R_{i,j}]_{ab}$ is 1 if the b^{th} entry of the image M appears in the a^{th} entry of the (vectorized) patch $P_{i,j}$, and 0 otherwise. For example, if $M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$ is a 3×3 image and patch $P_{2,2}$ is the 2×2 bottom right corner, then

$$R_{2,2}(M) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ m_{12} \\ m_{22} \\ m_{32} \\ m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = \begin{bmatrix} m_{22} \\ m_{32} \\ m_{23} \\ m_{33} \end{bmatrix} = P_{2,2} \sim \begin{bmatrix} m_{22} & m_{23} \\ m_{32} & m_{33} \end{bmatrix},$$

where $\sim$ represents reshaping a vectorized form to a matrix. It is now straightforward to write a MATLAB code `buildRij([N1 N2],[n1 n2],[s t])` to construct a matrix $R_{i,j}$ which operates on a vectorized $N_1 \times N_2$ image M to produce the vectorized $n_1 \times n_2$ patch whose upper-left index in M is (s,t).

For a collection $\mathcal{P}$ of patches of an image M, define $T_{\mathcal{P}} = \sum_{(i,j)\in\mathcal{P}} R_{i,j}^T R_{i,j}$.

Lemma 3.1 *For all $P_{i,j} \in \mathcal{P}$, $R_{i,j}^T R_{i,j}$ is diagonal and $T_{\mathcal{P}}$ is invertible. If the patches are non-overlapping, then $T_{\mathcal{P}}$ is the identity matrix.*

Proof. Let $P_{i,j}$ be some patch of M, both in vector form. Then $[R_{i,j}]_{rc} = 1$ iff $M_c = [P_{i,j}]_r$. Because $[P_{i,j}]_r$ must contain exactly one element of M, each row of $R_{i,j}$ contains exactly 1 nonzero entry. As each element of M appears at most once in $P_{i,j}$, each column of $R_{i,j}$ contains at most one nonzero element. This means that if $a \neq b$ then columns a and b of $R_{i,j}$ cannot have a nonzero element in the same row, so $[R_{i,j}^T R_{i,j}]_{a,b}$ (being the dot product of columns a and b of $R_{i,j}$) must be zero. So $R_{i,j}^T R_{i,j}$ is diagonal. Along the diagonal, $[R_{i,j}^T R_{i,j}]_{a,a}$ is the dot product of the a^{th} column of $R_{i,j}$ with itself, and is thus 1 iff the a^{th} entry of M appears in patch $P_{i,j}$. Since each entry of M appears in at least one patch, it follows that summing $R_{i,j}^T R_{i,j}$ over all patch indices (i,j) ensures $T_{\mathcal{P}}$ has nonzero diagonals, and is therefore invertible. When the patches are non-overlapping, each entry of M appears in only one patch, so $[R_{i,j} R_{i,j}^T]_{a,a} = 1$ for all $a = 1, \ldots, N_1 N_2$, by which we find $T_{\mathcal{P}}$ to be the identity matrix. □

We then express the image M as

$$M = (T_{\mathcal{P}})^{-1} \left(\sum_{(i,j)\in\mathcal{P}} R_{i,j}^T R_{i,j}(M) \right).$$

Because in this paper we use only non-overlapping patches, $T_{\mathcal{P}}$ is the identity, thus

$$M = \sum_{(i,j)\in\mathcal{P}} R_{i,j}^T R_{i,j}(M).$$

For any $(i,j) \in \mathcal{P}$, let $y_{i,j} \in \mathbb{R}^K$ be a sparse representation of the patch $R_{i,j}(M) = P_{i,j}$ under an $n_1 n_2 \times K$ dictionary D, so that $P_{i,j} = D y_{i,j}$. We thus say

$$M = \sum_{(i,j)\in\mathcal{P}} R_{i,j}^T D y_{i,j}.$$

To reconstruct the image with sparsely represented patches in a way which fits the observed data, we solve the following:

$$\min_{\substack{y_{i,j} \\ (i,j)\in\mathcal{P}}} \quad \sum_{(i,j)\in\mathcal{P}} \|y_{i,j}\|_1 - \sum_{(i,j)\in\mathcal{P}} \|y_{i,j}\| + \frac{\nu}{2} \left\| A \left(\sum_{(i,j)\in\mathcal{P}} R_{i,j}^T D y_{i,j} \right) - b \right\|^2, \tag{3.1}$$

and immediately note that (3.1) can be expressed as

$$\min_{y} \quad \frac{\nu}{2} \|\mathcal{A}y - b\|^2 + \|y\|_1 - \|y\|, \tag{3.2}$$

where $y = \begin{bmatrix} y_{11}^T & y_{12}^T & \cdots & y_{\bar{i}\bar{j}}^T \end{bmatrix}^T$ is the concatenation of all the sparse representations of patches under D, $\bar{i}$ and $\bar{j}$ the final row and column, respectively, of the patch partition, and finally

$$\mathcal{A} = A \begin{bmatrix} R_{11}^T D & R_{12}^T D & \ldots & R_{\bar{i}\bar{j}}^T D \end{bmatrix}.$$

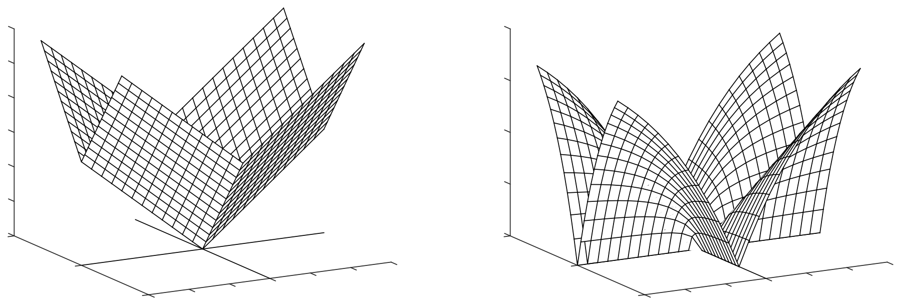


Figure 2: Surface plots of $\|x\|_1$ *(left)* and $\|x\|_1 - \|x\|$ *(right)* on $\mathbb{R}^2$. Minimizing an objective function containing these terms drives solutions towards the axes, simulating sparsity. Note that $\ell_1 - \ell_2$ regularization is not convex.

3.2 DCA for $\ell_1 - \ell_2$ Regularization

Use of $(\ell_1 - \ell_2)$ regularization causes the objective function to no longer convex, and so we adopt the Difference of Convex Functions Algorithm (DCA), used to minimize $g - h$ where $g\colon \mathbb{R}^n \to \mathbb{R}$ and $h\colon \mathbb{R}^n \to \mathbb{R}$ are both convex. The algorithm, developed by Tao and An in [10] and [11] is as follows:

DCA Algorithm

INPUT: x_1, $N \in \mathbb{N}$

for $k = 1, \dots, N$ **do**

  Find $y_k \in \partial h(x_k)$

  Find $x_{k+1} \in \partial g^*(y_k)$

end for

OUTPUT: x_{N+1}

Before using the DCA, we first apply Nesterov's smoothing from Corollary 2.14. Given a function of the form

$$f_0(x) = \max_{u \in Q}\{\langle Ax, u\rangle - \psi(u)\}$$

where $Q \subset \mathbb{R}^m$ is a convex, closed, and bounded set, ψ is a convex map from $\mathbb{R}^m$ to $\mathbb{R}$, and A is an $m \times n$ matrix, for any $\mu > 0$ we may obtain a smooth approximation f_μ using

$$f_\mu(x) = \max_{u \in Q}\{\langle Ax, u\rangle - \psi(u) - \frac{\mu}{2}\|u\|^2\}.$$

We now use this to smooth the ℓ_1 norm. Let $p(x) = \|x\|_1$, and note that this is equivalent to $p(x) = \max_{u \in Q}\{\langle x, u\rangle\}$ when Q is the unit box, $Q = \{x \in \mathbb{R}^n \mid |x_i| \le 1\,,\, i = 1, \dots, n\}$. In the above general setting, this corresponds to A being the identity matrix and ψ the zero map. Then $p_\mu(x)$ is a smooth approximation to $p(x) = \|x\|_1 = \max_{u\in Q}\{\langle x, u\rangle\}$ and can be expressed

$$p_\mu(x) = \frac{1}{2\mu}\|x\|^2 - \frac{\mu}{2}\left(d\left(\frac{x}{\mu}, Q\right)\right)^2,$$

where $d(x;Q)$ is the Euclidean distance from x to Q.

Now let A be a real $m \times n$ matrix and $b \in \mathbb{R}^m$. Using the above smooth approximation for $\|x\|_1$ we approximate $f(x) = \frac{\nu}{2}\|Ax-b\|^2 + \|x\|_1 - \|x\|$ with

$$\begin{aligned} f_\mu(x) &= \frac{\nu}{2}\|Ax-b\|^2 + \frac{1}{2\mu}\|x\|^2 - \frac{\mu}{2}\left(d\left(\mu^{-1}x, Q\right)\right)^2 - \|x\| \\ &= \frac{1}{2\mu}\|x\|^2 - \left(\frac{\mu}{2}\left(d\left(\mu^{-1}x, Q\right)\right)^2 - \frac{\nu}{2}\|Ax-b\|^2 + \|x\|\right) \\ &= \frac{1}{2\mu}\|x\|^2 + \frac{\gamma}{2}\|x\|^2 - \left(\frac{\mu}{2}\left(d\left(\mu^{-1}x, Q\right)\right)^2 - \frac{\nu}{2}\|Ax-b\|^2 + \frac{\gamma}{2}\|x\|^2 + \|x\|\right). \end{aligned}$$

Note this is the difference of convex functions $g - h$ for

$$g(x) = \left(\frac{1+\mu\gamma}{2\mu}\right)\|x\|^2 \text{ and } h(x) = \frac{\mu}{2}\left(d\left(\mu^{-1}x, Q\right)\right)^2 - \frac{\nu}{2}\|Ax-b\|^2 + \frac{\gamma}{2}\|x\|^2 + \|x\|,$$

assuming that $\gamma > 0$ is sufficiently large so that $\frac{\gamma}{2}\|x\|^2 - \frac{\nu}{2}\|Ax-b\|^2$ is convex. Note this is satisfied when γ is greater than ν times the largest eigenvalue of A^TA.

To use the DCA algorithm, we will need y_k in the subdifferential of h at x_k. Using

$$\nabla\|Ax-b\|^2 = 2A^T(Ax-b)$$

and

$$\nabla\left(d\left(x, Q\right)\right)^2 = 2(x - \Pi_Q(x))$$

where $\Pi_Q(x)$ is the projection of x onto Q, along with the chain rule for subdifferentials (see [7]), we have a subgradient of h at x given by

$$\partial_w h(x) = \mu^{-1}x - \Pi_Q\left(\mu^{-1}x\right) - \nu A^T(Ax-b) + \gamma x + \omega(x),$$

where $\omega(x) = \begin{cases} \frac{x}{\|x\|} & x \neq 0 \\ 0 & x = 0 \end{cases}$ is a subgradient of $\|\cdot\|$ at x. We point out that the projection onto the unit box can be defined component-wise as $[\Pi_Q(x)]_i = \max(-1, \min(x_i, 1))$.

To find $x_{k+1} \in \partial^* g(y_k)$, we use the fact that $u \in \partial^* g(v)$ iff $v \in \partial g(u)$. The subdifferential of g is simply the singleton set containing its gradient (see [7]), so $v \in \partial g(u)$ iff $v = \frac{1+\mu\gamma}{\mu}u$ iff $u = \frac{\mu}{1+\mu\gamma}v$.

We combine these results to implement the DCA algorithm in order to minimize a μ-smoothing approximation to $f(x) = \frac{\nu}{2}\|Ax-b\|^2 + \|x\|_1 - \|x\|$, as outlined below.

Algorithm 1. DCA for smoothed $\ell_1 - \ell_2$ regularization.
INPUT: $\mu > 0$, sufficiently large γ, starting point x
repeat
Find $\omega = \frac{x}{\|x\|}$ if $x \neq 0$, $\omega = 0$ otherwise
$y \longleftarrow \mu^{-1}x - \Pi_Q\left(\mu^{-1}x\right) - \nu A^\top(Ax - b) + \gamma x + \omega$
$x \longleftarrow \frac{\mu}{1+\mu\gamma}y$
until convergence
OUTPUT: x

Experiments suggest that incrementally decreasing μ over the course of the algorithm induces better performance.

3.3 Choosing Partitions

This section describes how to obtain t different partitions of the image, following the approach described in [15]. Given an $N_1 \times N_2$ image matrix, choose a general patch size $n_1 \times n_2$. We then choose a size $c_1 \times c_2$ of the upper- and left-most patch, P_{11}, where $c_i \leq n_i$ for $i = 1, 2$. All of the patches not on the boundary of the image will have size $n_1 \times n_2$. The left-boundary non-corner patches of M are size $n_1 \times c_2$, the upper-boundary non-corner patches have size $c_1 \times n_2$, and the remaining patch sizes are chosen to ensure their borders align with those patches already defined.

Figure 3: Two partitions of an $(N_1 \times N_2) = (4 \times 5)$ image with patch size $(n_1, n_2) = (2, 2)$. Left: The top-left corner has size $(c_1, c_2) = (2, 2)$. Right: The top-left patch has size $(c_1, c_2) = (1, 1)$.

If patch $P_{i,j}$ has size less than $n_1 \times n_2$, the patch extraction operator $R_{i,j}$ still creates a patch of size $n_1 \times n_2$, in which $P_{i,j}$ sits in the proper orientation, and the remaining entries are zeros. Similarly, $R_{i,j}^T$ will embed an $n_1 \times n_2$ patch into the corresponding patch in the image, but zero out all entries which do not lie in the smaller patch. For example, if the general patch size is 8×8 but the corner patch P_{11} is 5×5, then R_{11} embeds the top left 5×5 patch into an 8×8 patch of zeros. We say P_{11} has 'virtual size' 5×5. Note that the cell array of patch extraction matrices does not need to be constructed every time a problem is solved. Once it has been constructed for some partition of a given size image, it can be saved and reused. The general algorithm given in [15] is as follows: Given a dictionary D, choose some t different patch-partitions of the image, $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_t$. For each $k = 1, \ldots, t$, find the solution M_k to the unconstrained problem (3.2) using partition $\mathcal{P}_k$. Then use the average of those solutions, $\overline{M} = \frac{1}{t}\sum M_k$, as the final reconstruction.

3.4 Dictionaries

In this summary we use two types of dictionary. One is constructed from the discrete cosine transform (DCT). The other is a 'learned dictionary,' constructed using a collection of images as training data, and for which the learned dictionary allows sparse representations. The i, j entry of an $M \times N$ discrete cosine transform (DCT-II) matrix D is given by

$$D_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} & j = 1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi}{N}(j-1)(i+\frac{1}{2})\right) & j = 2, \ldots, N \end{cases}.$$

Alternatively, a 'wavelet' dictionary can be called using MATLAB's `wmpdictionary( )` function, with argument equal to the number of atoms.

We find better results when we 'learn' a dictionary from a training data. Consider a training matrix $X = [x_1, \ldots, x_L] \in \mathbb{R}^{n \times L}$ of L images of size n in vectorized form. We seek a dictionary $D = [d_1, \ldots, d_K] \in \mathbb{R}^{n \times K}$ of K atoms of size n and a corresponding coefficient matrix $W = [w_1, \ldots, w_L] \in \mathbb{R}^{K \times L}$ so that $x_i \approx Dw_i$ and w_i is as sparse as possible, for all $i = 1, \ldots, L$.

There exist several methods for learning a dictionary. One of the most popular algorithms is the K-SVD proposed in [1] which can be modeled as

$$\begin{aligned} \min_{D,W} \quad & \|DW - X\|_F^2 \\ \text{subject to} \quad & \|d_i\| = 1 \text{ for all } i = 1, \ldots, K \text{ and } \|w_j\|_0 \leq s \text{ for all } j = 1, \ldots, L, \end{aligned}$$

where s is a parameter to control the sparsity. Another popular method is the Online Dictionary Learning (OLM) proposed in [5] which solves the following problem:

$$\begin{aligned} \min_{D,W} \quad & \frac{\lambda}{2}\|DW - X\|_F^2 + \|W\|_1 \\ \text{subject to} \quad & \|d_i\| = 1 \text{ for all } i = 1, \ldots, K, \end{aligned} \tag{3.3}$$

where $\|W\|_1 = \sum_{i=1}^L \|w_i\|_1 = \sum_{i=1}^L \sum_{j=1}^K |w_{ij}|$ and λ is a trade-off parameter to balance data fitting and sparsity level.

To promote the sparsity, our approach is to use the $\ell_1 - \ell_2$ regularization by solving the following problem:

$$\begin{aligned} \min_{D,W} \quad & \frac{\lambda}{2}\|DW - X\|_F^2 + \|W\|_1 - \|W\|_{2,1} \\ \text{subject to} \quad & \|d_i\| \leq 1 \text{ for all } i = 1, \ldots, K, \end{aligned} \tag{3.4}$$

where $\|W\|_{2,1} = \sum_{i=1}^L \|w_i\| = \sum_{i=1}^L \sqrt{\sum_{j=1}^K w_{ji}^2}$. This is a nonconvex problem whose nonconvexity comes from two sources: the sparsity promotion $\ell_1 - \ell_2$ and the bi-linearity between the dictionary D and the code W in the fitting term.

For solving this problem, we alternatively update W and D by using the DCA and Nesterov's smoothing.

1. **Sparse coding phase**: In this phase, we fix a dictionary D and try to update the code W by solving (3.4). The objective function is now a DC function with respect to W:

$$f(W) = \frac{\lambda}{2}\|DW - X\|_F^2 + \|W\|_1 - \|W\|_{2,1}.$$

Let $P(W) = \|W\|_1$. Using the smoothing technique as before, we can approximate the function $P(W)$ by

$$P_\mu(W) = \sum_{i=1}^{L}\left[\frac{1}{2\mu}\|w_i\|^2 - \frac{\mu}{2}[d(\frac{w_i}{\mu};Q)]^2\right] = \frac{1}{2\mu}\|W\|_F^2 - \frac{\mu}{2}\sum_{i=1}^{L}[d(\frac{w_i}{\mu};Q)]^2,$$

where $Q = \{w \in \mathbb{R}^K \mid \|w\|_\infty \leq 1\}$. Recall that the ith component of the Euclidean projection from $w \in \mathbb{R}^K$ onto the box Q can be computed as

$$[\Pi_Q(w)]_i = \max(-1, \min(1, w_i)). \tag{3.5}$$

To process further, we denote $\mathcal{Q} = Q \times Q \times \ldots \times Q \subset \mathbb{R}^{K\times L}$. For an $K \times L$ matrix W, the projection from $W = [w_1, \ldots, w_L]$ onto $\mathcal{Q}$ is define by

$$\Pi\left(W, \mathcal{Q}\right) = [\Pi(w_1; Q), \ldots, \Pi(w_L; Q)] \in \mathbb{R}^{K\times L}.$$

We thus have

$$[d(W;\mathcal{Q})]^2 = \|W - \Pi(W,\mathcal{Q})\|_F^2 = \sum_{i=1}^{L}[d(w_i;Q)]^2.$$

The function $f(W)$ can be approximated by the DC function $f_\mu(W) = g_\mu(W) - h_\mu(W)$, where

$$g_\mu(W) = \left(\frac{1}{2\mu} + \frac{\gamma_1}{2}\right)\|W\|_F^2,$$
$$h_\mu(W) = \frac{\mu}{2}[d(\frac{W}{\mu};\mathcal{Q})]^2 - \frac{\lambda}{2}\|DW - X\|_F^2 + \frac{\gamma_1}{2}\|W\|_F^2 + \|W\|_{2,1},$$

and γ_1 is chosen such that $\frac{\gamma_1}{\lambda}$ is greater than the spectral radius of the symmetric matrix $D^\top D$ in order to guarantee the convexity of the function $h_\mu(W)$.

A subgradient Y of h_μ at W is given by

$$Y = \frac{W}{\mu} - \Pi\left(\frac{W}{\mu};\mathcal{Q}\right) - \lambda D^\top(DW - X) + \gamma_1 W + \eta(W),$$

where $\eta(W)$ is an $K \times L$ matrix whose i^{th} column is defined via the i^{th} column of W by

$$[\eta(W)]_i = \begin{cases} \frac{w_i}{\|w_i\|} & \text{if } w_i \neq 0, \\ 0_{\mathbb{R}^K} & \text{if } w_i = 0. \end{cases} \tag{3.6}$$

The DCA for solving the sparse coding phase can be outlined as follows.

Algorithm 2. DCA for sparse coding phase.

INPUT: $X \in \mathbb{R}^{n\times L}$, $D \in \mathbb{R}^{n\times K}$, $\mu > 0$, $\lambda > 0$ sufficiently small, $\gamma_1 > 0$ sufficiently large and starting code $W \in \mathbb{R}^{K\times L}$.

repeat

Find $\Pi(W, \mathcal{Q}) = [\Pi(w_1; Q), \ldots, \Pi(w_L; Q)]$ according to (3.5)

Find $\eta(W)$ according to (3.6)

$Y \longleftarrow \frac{W}{\mu} - \Pi\left(\frac{W}{\mu}; \mathcal{Q}\right) - \lambda D^\top(DW - X) + \gamma_1 W + \eta(W)$

$W \longleftarrow \dfrac{\mu}{1+\mu\gamma_1} Y$

until convergence

OUTPUT: W

2. **Dictionary updating phase.** Now we fix the sparse code W that has been found from the previous phase and update the dictionary D by solving

$$\min_D \quad \|DW - X\|_F^2 \qquad \text{subject to} \qquad \|d_i\| \le 1 \text{ for all } i = 1, \ldots, K.$$

For solving this nonconvex problem, we use the DCA by reformulating it as a DC programming problem as follows

$$\min_D \qquad \widetilde{f}(D) = \left[\frac{\gamma_2}{2}\|D\|_F^2 + I_{\mathcal{C}}(D)\right] - \left[\frac{\gamma_2}{2}\|D\|_F^2 - \|DW - X\|_F^2\right],$$

where $\mathcal{C} = \{D = [d_1, \ldots, d_K] \in \mathbb{R}^{n\times K} \mid \|d_i\| \le 1 \text{ for all } i = 1, \ldots, K\}$ is the constraint. Here γ_2 is chosen greater than the spectral radius of the matrix $WW^\top$ to ensure the convexity of the function $\widetilde{h}(D) = \frac{\gamma_2}{2}\|D\|_F^2 - \|DW - X\|_F^2$.

This function $\widetilde{h}$ is differentiable and its gradient given by

$$\nabla\widetilde{h}(D) = \gamma_2 D - [DW - X]W^\top.$$

Note that the ith component of the Euclidean projection from D onto the constraint $\mathcal{C}$ can be computed by

$$[\Pi_{\mathcal{C}}(D)]_i = \frac{d_i}{\max\{1, \|d_i\|\}}, \text{ for } i = 1, \ldots, K.$$

Thus, the DCA iterative sequence in this phase is simply defined by $D_{k+1} = \Pi_{\mathcal{C}}\left(\dfrac{\nabla h(D_k)}{\gamma_2}\right)$.

Algorithm 3. DCA for dictionary updating phase.

INPUT: $X \in \mathbb{R}^{n\times L}$, $W \in \mathbb{R}^{K\times L}$, $\gamma_2 > 0$ sufficiently large, starting dictionary $D \in \mathbb{R}^{n\times K}$.

repeat

$Y \longleftarrow \gamma_2 D - [DW - X]W^\top$

$D \longleftarrow \Pi_{\mathcal{C}}\left(\frac{Y}{\gamma_2}\right)$

until convergence

OUTPUT: D

In practice, when alternatively perform Algorithm 2 and Algorithm 3 to solve (3.4), we can use a value $\gamma > 0$ sufficiently large to play the role of both γ_1 and γ_2. In addition, we also gradually decrease the value of smoothing parameter μ until a preferred μ_∞ is attained. The final scheme for $\ell_1 - \ell_2$ dictionary learning can be outlined as follows.

Algorithm 4. DCA for $\ell_1 - \ell_2$ dictionary learning.

INPUT: training set $X \in \mathbb{R}^{n\times L}$, $\lambda > 0$ sufficiently small, $\gamma > 0$ sufficiently large,
starting dictionary $D^0 \in \mathbb{R}^{n\times K}$, starting code $W^0 \in \mathbb{R}^{K\times L}$
$\mu_0 > 0$, $\sigma \in (0, 1)$ and μ_∞ sufficiently small.

$k \longleftarrow 0$

repeat

Compute $W^{k+1} \longleftarrow$ **Algorithm 2**$(X, D^k, W^k, \lambda, \gamma, \mu_k)$

Compute $D^{k+1} \longleftarrow$ **Algorithm 3**$(X, D^k, W^{k+1}, \lambda, \gamma, \mu_k)$

Update $\mu_{k+1} \longleftarrow \sigma\mu_k$

Set $k \longleftarrow k + 1$

until $\mu < \mu_\infty$.

OUTPUT: D

3.5 Implementation

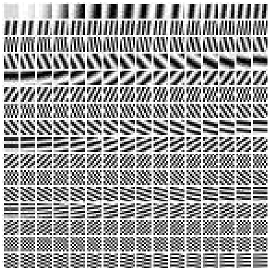
Our goal is to restore an unknown image M of size $N_1 \times N_2$ from its corrupted linear measurements of the form $b = A(M) + \varepsilon$. We first choose a general patch size $n_1 \times n_2$ with $n_i \ll N_i$ for $i = 1, 2$. Then we generate a dictionary of size $n_1n_2 \times K$ by using DCT or learning from a training data set X of size $n_1n_2 \times L$ with $n_1n_2 \le K \ll L$. Let $\mathcal{P}$ be a patch partition associated with some choice of upper-left-most patch and let S be the number of patches in $\mathcal{P}$. For any $(i, j) \in \{(1, 1), \ldots, (\bar{i}, \bar{j})\}$, we find the extraction operator R_{ij} and form the matrices

$$R = \begin{bmatrix} R_{11} \\ R_{12} \\ \vdots \\ R_{\bar{i}\bar{j}} \end{bmatrix} \qquad \text{and} \qquad R^T = \begin{bmatrix} R_{11}^T & R_{12}^T & \cdots & R_{\bar{i}\bar{j}}^T \end{bmatrix}.$$

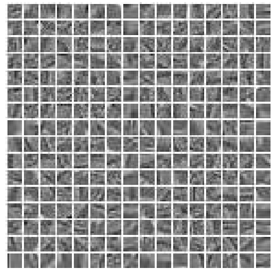
We continue by solving (3.2) to find $y \in \mathbb{R}^{KS}$. Then express $y \in \mathbb{R}^{KS}$ as an $K \times S$ matrix Y of patch representations under D, so $Y = [y_{11}\ y_{12}\ \cdots\ y_{\bar{i}\bar{j}}]$ and DY is an $n_1n_2 \times S$ matrix whose columns are vectorized patches. Finally, reshaping DY into $n_1n_2S \times 1$ vectorized form $\overline{Dy}$, we have $R^T\overline{Dy} = \sum_{i,j} R_{i,j}^T Dy_{i,j}$ is an $N_1 \times N_2$ image in vectorized form.

We now use the above scheme to solve in-painting problems, where A is the sampling operator. In-painting is a process wherein missing information in an image is recovered, namely when some known set of pixels of an image have been lost. Let $M \in \mathbb{R}^{N_1N_2}$ be a vectorization of an $N_1 \times N_2$ image, Ω a subset of $\{1, \ldots, N_1N_2\}$ and A be the $|\Omega| \times N_1N_2$ matrix formed by removing all row i from the identity matrix $I_{N_1N_2}$ for all $i \notin \Omega$. Then we call A a sampling operator with sampling rate $\mathsf{SR} = \frac{|\Omega|}{N_1N_2}$, and $A(M)$ is a vectorization of the original image, containing only those pixels indexed by Ω.

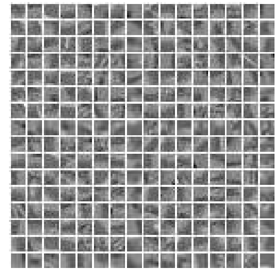
The patching approach developed by Xu and Yin [15] is implemented to minimize (3.2) using the DCA with Nesterov's smoothing. In all settings we compare the discrete cosine transform (DCT) dictionary with two different learned dictionaries: ℓ_1 regularization by solving (3.3) with block proximal gradient (BPG) proposed in [14, 15] and $\ell_1 - \ell_2$ regularization by minimizing (3.4) with Algorithm 4.



DCT dictionary



ℓ_1 learned dictionary by BPG



$\ell_1 - \ell_2$ learned dictionary by Alg 4

Figure 4: Three different types of dictionaries

SR			Rel. Error (%)	PSNR
50%		corrupted image	70.72	8.458
	DCT	FISTA	**4.81**	**31.83**
		DCA	6.06	29.81
	ℓ_1 *learned*	FISTA	**3.45**	**34.699**
		DCA	4.11	33.168
	$\ell_1 - \ell_2$ *learned*	FISTA	**3.48**	**34.613**
		DCA	4.22	33.937
30%		corrupted image	83.77	6.987
	DCT	FISTA	**7.01**	**28.501**
		DCA	8.27	27.101
	ℓ_1 *learned*	FISTA	**5.21**	**31.113**
		DCA	5.89	30.048
	$\ell_1 - \ell_2$ *learned*	FISTA	**5.02**	**31.438**
		DCA	5.72	30.303

Table 1: Results for in-painting with three different dictionaries. FISTA and DCA are ℓ_1 and $(\ell_1 - \ell_2)$ regularization, respectively. Best results are in bold.

For all learned dictionaries, we use a training set of 10000 grayscale patches of size 8×8, chosen randomly from 100 images taken from the Berkeley Segmentation Dataset[5]; see [6]. The training matrix X is of size 64×10000. The number of atoms for learned dictionaries is set to be $K = 256$ and thus all learned dictionaries are of size 64×256.

A technical step before performing the DCA-based learning algorithm is to set each column of the training matrix X to zero mean. For the ℓ_1 regularization, we solve (3.3) with $\lambda = 0.1$ by the BPG method using the same parameters as in [15, Algorithm 3]. For $\ell_1 - \ell_2$ regularization, we randomly choose K columns from the training matrix X and normalize them to form a starting dictionary D^0 when solving (3.4) by Algorithm 4 with $W^0 = \texttt{pinv}(D)X$, $\lambda = 1$, $\gamma = 2000$, $\sigma = 0.8$, $\mu_\infty = 10^{-5}$. The obtained dictionaries are shown in Figure 4.

[5]available at `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`

Sampled image (SR=50%)

FISTA, DCT dictionary

DCA, DCT dictionary

FISTA, ℓ_1 learned dictionary

DCA, ℓ_1 learned dictionary

FISTA, $\ell_1 - \ell_2$ learned dictionary

DCA, $\ell_1 - \ell_2$ learned dictionary

Figure 5: In-painting result on *Lena512* with DCT and learned dictionaries. FISTA and DCA are ℓ_1 and $(\ell_1 - \ell_2)$ regularization, respectively.

For all tests, we use the 512×512 standard reference image *Lena*, and choose $n_1 \times n_2 = 8 \times 8$ patches. Before running the test, a column of all ones is added to the DCT and learned dictionaries. As discussed in [15], patching artifacts which appear in the solution are mitigated by processing the image three times, each with a different partition. The solution is then taken to be the average of the three trials. Our partitions were determined by choosing upper-left corner patches of size 8×8, 5×5, and 2×2.

Corrupted images were defined as $b = A(M) + \sigma\xi$, where ξ is a matrix of noise with standard normal distribution scaled by $\sigma = c\frac{\|A(M)\|_2}{\|\xi\|_2}$.

In our experiments, we fix the noise level $c = 1\%$ and use $\nu = \frac{1}{2\sigma}$ for ℓ_1 regularization with FISTA [2] and $\nu = \frac{3}{20\sigma}$ for $\ell_1 - \ell_2$ regularization with Algorithm 1.

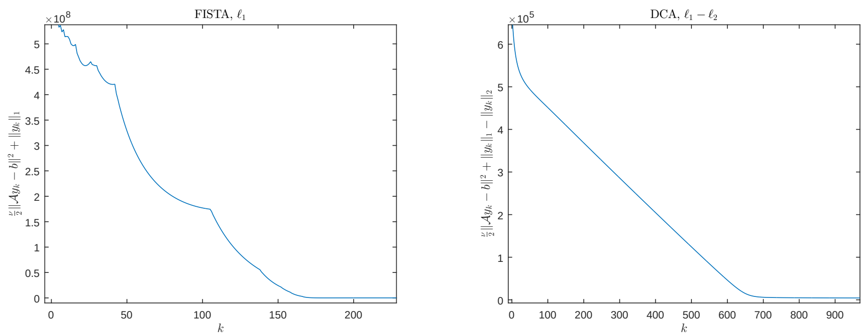


Figure 6: Inpainting: objective function value vs iteration for FISTA and DCA with learned dictionary.

We measure error of the solution $\widetilde{M}$ relative to the true image M by relative error, $\mathsf{RE} = \frac{\|\mathsf{M}-\widetilde{\mathsf{M}}\|_\mathsf{F}}{\|\mathsf{M}\|_\mathsf{F}}$, and peak signal to noise ratio as $\mathsf{PSNR} = 20 \cdot \log_{10}\left(\frac{\sqrt{\mathsf{N_1N_2}}}{\|\mathsf{M}-\widetilde{\mathsf{M}}\|_\mathsf{F}}\right)$. See Table 1 for a comparison of the PSNR values and relative errors of the in-paiting result with different sampling rates and different dictionaries. Figure 5 gives a visual illustration for the case $\mathsf{SR} = 50\%$. Given these results, it is evident that $\ell_1 - \ell_2$ learned dictionary obtained from Algorithm 4 yields results very close to the one constructed by BPG method. Moreover, it can be seen that the performance of DCA with smoothing technique is nearly comparable to that of the FISTA on learned dictionaries.

3.6 Discussion

The fast patch dictionary method given by Xu and Yin [15] was qualitatively successful in reconstructing corrupted images, using both ℓ_1 regularization with FISTA, and $(\ell_1 - \ell_2)$ regularization with DCA in combination with Nesterov's smoothing. In every case, learned dictionaries improve results compared to a DCT dictionary.

The FISTA approach converges after fewer iterations (see Figures 6), but DCA required less time per iteration. The optimal choice of μ and γ parameters in the DCA method is unknown, and allows for the possibility of future improvement. Similarly, implementing FISTA without a backtracking line search is likely to induce better results, in cases where the Lipschitz constant of the gradient can be determined. Also, it is not known which choice of ν (used to weight data-fitting versus sparsity) leads to the best solution. Future work may explore optimal parameter choice as well as characterize which problems benefit from ℓ_1 versus $(\ell_1 - \ell_2)$ regularization.

Acknowledgements. Part of this work was done during the first author's visit to the Vietnam Institute for Advanced Study in Mathematics (VIASM). He would like to thank the VIASM and Prof. Nguyen Dong Yen for the hospitality and support.

References

[1] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54** (2006), 4311–4322.

[2] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2** (2009), 183–202.

[3] F. H. Clarke, Nonsmooth Analysis and Optimization, John Wiley & Sons, Inc., New York, 1983.

[4] J. R. Giles, A Survey of Clarkes Subdifferential and the Differentiability of Locally Lipschitz Functions. In: Progress in Optimization. Applied Optimization, vol 30. Springer, Boston, MA.

[5] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding. Proc. 26th Int'l Conf. Machine Learning. Montreal, Canada, 2009.

[6] D. Martin, C. Fowlkes, D. Tal, and J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proc. 8th Int'l Conf. Computer Vision. **2** (2001), 416–423.

[7] B.S. Mordukhovich and N.M. Nam, An Easy Path to Convex Analysis and Applications, Morgan & Claypool, 2014.

[8] N.M. Nam, N.T. An, R.B. Rector, and J. Sun, Nonsmooth algorithms and Nesterov smoothing techniques for generalized Fermat-Torricelli problems. SIAM J. Optim. **24** (2014), 1815–1839.

[9] Y. Nesterov, Smooth minimization of non-smooth functions. Math.Program., Ser. A, **103** (2005), 127–152.

[10] P.D. Tao, L.T.H. An, Convex analysis approach to D.C. programming: Theory, algorithms and applications. Acta Math. Vietnam. **22** (1997), 289–355.

[11] P.D. Tao, L.T.H. An, A d.c. optimization algorithm for solving the trust-region subproblem. SIAM J. Optim., **8** (1998), 476–505.

[12] J. Xin, S. Osher, and Y. Lou, Computational aspects of L1-L2 minimization for compressive sensing. Advances in Intelligent Systems and Computing, **359** (2015), 169–180.

[13] P. Yin, Y. Lou, Qi He, and J. Xin, Minimization of L1-L2 for compressed sensing. SIAM J. of Sci. Comput. **37** (2015), A536–A563.

[14] Y. Xu and W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM SIAM J. Imaging Sci. **6**(2013), 1758–1789.

[15] Y. Xu and W. Yin, A fast patch dictionary method for whole image recovery. Inverse Problems and Imaging, **10** (2016), 563–583.