A DIRECT DERIVATION FOR THE L-INFINITY-NORM PROXIMAL OPERATOR

Ender Erkaya, Alper T. Erdogan

Department of Electrical and Electronics Engineering
Koc University
Istanbul, Turkey

ABSTRACT

ℓ_∞-norm is an important component of optimization settings put forward for various applications. In this article, the proximal operator for ℓ_∞-norm is derived based directly on the optimality condition of the corresponding cost function. It is shown that the proximal operator is simply equivalent to an element-wise clipping operation, where the clipping level is dependent on the argument of the proximal operator. The article also provides a simplified procedure for obtaining the clipping level. The numerical example provided illustrates the potential speed and performance improvement obtained by the proposed proximal operator.

Index Terms— ℓ_∞-norm, proximal algorithms.

1. INTRODUCTION

Optimization settings featuring ℓ_∞-norm appear in various application areas: they naturally arise in min-max criterion based approaches such as frequency sampling based filter design and minimum sidelobe beamforming. Peak-to-average power ratio (PAPR) minimization for multicarrier (OFDM/DMT) communication systems is also a prototype ℓ_∞-norm minimization problem.

Inverse problems is another area where ℓ_∞-norm has been proven to be useful. Based on its duality with ℓ_1-norm , the minimization of the ℓ_∞-norm of the output of an LTI system, leads to sparsification of the corresponding impulse response, under some constraints on the inputs and inverse system parameters. This feature was successfully exploited in blind equalization domain to derive convex cost function based adaptive algorithms [1, 2, 3, 4], and later extended to Blind Source Separation (BSS) domain leading the development of Bounded Component Analysis algorithms [5, 6]. More recently, the *compressed training* approach has been proposed which effectively uses ℓ_∞-norm as a regularizer for the least squares based training setup leading to logarithmic decrease in the training length requirements of the communication systems [7, 8].

As the use of ℓ_∞-norm has spread to wide range of applications, the need for fast and low complexity algorithms to solve such problems has also become critical. From this perspective, the convexity of ℓ_∞ provides us with the opportunity to search among wide range algorithms studied in the convex optimization area for several decades. However, its nondifferentiability eliminates some algorithms such as gradient descent and Newton's method. Although most problems involving ℓ_∞ can be cast as a second order cone programming and solved via interior point methods, large data sizes and dense matrices annihilate advantages of the complex interior point algorithms that diminishes them as a fast and good option [9]. As an explicit approach, subgradient based methods can be used to solve ℓ_∞ norm based optimization problems by handling the nondifferentiability of the ℓ_∞ norm. However, slow convergence of subgradient methods push us to search for faster algorithms.

In the recent literature, many fast and low complexity algorithms based on proximal operators have been proposed for convex and non-differentiable problems, and successfully applied to especially ℓ_1-norm based cost functions arising in signal processing and machine learning applications. In order to adapt these algorithms to ℓ_∞ based optimization problem schemes, we primarily need to get a fast, low complexity and easy to implement proximal operator of the function $f = \|.\|_\infty$.

The ℓ_∞-norm operator can be derived based on Moreau identity[10] and considering the dual problem of projection to ℓ_1-norm ball. In reference [11], an efficient algorithm for projection to ℓ_1-norm ball is proposed, which is derived based on the projection to unit-simplex in [12]. In [13], the use of the ℓ_∞-norm proximal operator based on this approach was proposed. The main purpose of this article is to obtain the proximal operator for the ℓ_∞-norm using the optimality condition for the ℓ_∞-proximal operator cost function. We show that the ℓ_∞-norm proximal operator is simply equal to the clipping function. The proposed derivation is more direct and reflects the clipping function behavior in a clear manner. Furthermore, this derivation approach can be easily generalized to k-peak function, i.e. the sum of the k largest magnitude entries [14].

The article is organized as follows: in Section 2, we first remind the basic definition of the proximal operator. Section 3 is the main section, where the proximal operator for ℓ_∞-norm is derived. Finally, in Section 4, we provide a numerical example to illustrate the use of the proposed proximal operator.

2. PROXIMAL OPERATORS

The proximal operator for the convex function $f(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^n$ is defined as [10]

$$prox_{\lambda f}(\mathbf{v}) = \arg\min_{\mathbf{x}} \; f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x}-\mathbf{v}\|_2^2 . \tag{1}$$

$\mathbf{x}^*$ is the solution to the above optimization problem, output of the proximal operator, if and only if the optimality condition

$$\mathbf{0} \in \lambda \partial f(\mathbf{x}^*) + (\mathbf{x}^* - \mathbf{v}), \tag{2}$$

holds. Furthermore, $\mathbf{x}^*$ is the unique minimizer of the problem due to strong convexity coming from the quadratic part, $\|\mathbf{x}-\mathbf{v}\|_2^2$.

Rearranging the terms, the optimal condition turns to $\mathbf{x}^* = \mathbf{v} - \lambda\mathbf{s}$ where $\mathbf{s} \in \partial f(\mathbf{x}^*)$. In other words, we are essentially applying a subgradient algorithm step with stepsize λ where the subgradient is evaluated at output $\mathbf{x}^*$ on the contrary to the conventional subgradient method which uses the current subgradient information. Due to use of future(next step) subgradient information, proximal operator is considered as backward or implicit method in opposition to explicitness of (sub)gradient based methods [15].

3. ℓ_∞-NORM PROXIMAL OPERATOR

The following theorem shows that the proximal operator of the ℓ_∞-norm is equivalent to elementwise (absolute) clipping operation, where the clipping level is determined by λ:

Theorem: Consider the proximal operator $prox_{\lambda f}$ where $f = \|.\|_\infty$. The solution $\mathbf{x}^*$ to the problem

$$prox_{\lambda f}(\mathbf{v}) = \arg\min_{\mathbf{x}} \lambda \|\mathbf{x}\|_\infty + (1/2)\|\mathbf{x}-\mathbf{v}\|_2^2$$

is given by the elementwise clipping operator

$$\mathbf{x}^* \;=\; C_\mu(\mathbf{v}) = \begin{bmatrix} c_\mu(v_1) & c_\mu(v_2) & \dots & c_\mu(v_n) \end{bmatrix}^T$$

where

- $c_\mu(\cdot)$ is the scalar clipping function given in Fig. 1, which can be written as

$$c_\mu(q) = \begin{cases} -\mu & q < -\mu, \\ q & |q| \le \mu, \\ \mu & q > \mu. \end{cases} \tag{3}$$

- $\mu = \frac{1}{L}\{\mathbf{1}_L{}^T\mathbf{v}^{(s)} - \lambda\}^+$ is the clipping level,
- $\mathbf{v}^{(s)}$ is the vector obtained by taking the absolute values of the elements of $\mathbf{v}$ and sorting them in a descending order,

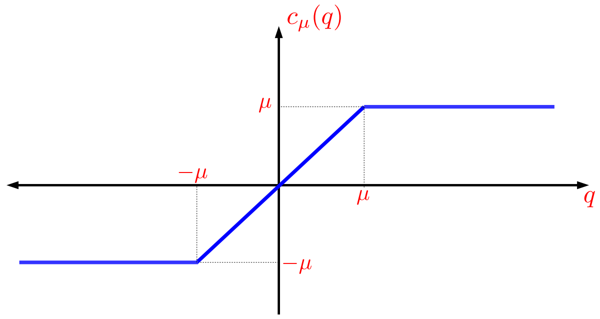

Fig. 1. Scalar clipping function.

- $\mathbf{1}_L \in \mathbb{R}^n$ is the partial ones vector defined as $1_L = [\underbrace{1, 1....., 1, 1}_{\text{L}} 0, 0....0, 0]^T$,
- $\{\mathbf{u}\}^+ = max(\mathbf{u}, \mathbf{0})$, is the elementwise Rectified Linear Unit (ReLU) operator,
- L is the the number of clipped elements. It can be determined as the smallest index with a nonnegative element of the vector $\mathbf{A}\mathbf{v}^{(s)} - \lambda\mathbf{1}$ if $\mathbf{A}\mathbf{v}^{(s)} - \lambda\mathbf{1} \notin \mathbb{R}^n_-$, otherwise $L = n$. Here, $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n\times n}$ is the index checking matrix defined by

$$A_{ij} = \begin{cases} 0 & i < j-1, \\ -\mathrm{i} & i = j-1, \\ 1 & i \ge j. \end{cases}$$

Proof: $\mathbf{x}^*$ is the solution, if it satisfies the optimality condition in (2) where the subdifferential set of f at $\mathbf{x} \in \mathbb{R}^n - \mathbf{0}$ is given by

$$\partial f(\mathbf{x}) = \{\mathbf{s} \in \mathbb{R}^n \mid \quad \mathbf{s} = \sum_{j\in I} \beta_j sign(x_j)\mathbf{e}_j \;\; ; \quad \sum_{j\in I} \beta_j = 1, \beta_j \in \mathbb{R}_+\}$$

where I is the index set of absolute maximum elements defined as $I = \{j \mid |x_j| = \|\mathbf{x}\|_\infty \;, j \in \{1, 2, ..n\}\}$ and $\mathbf{e}_j$ is the standard basis vector in $\mathbb{R}^n$ with all elements equal to zero except the j^{th} element is equal to 1. When $\mathbf{x} = \mathbf{0}$,

$$\partial f(\mathbf{0}) = \{\mathbf{s} \in \mathbb{R}^n \mid \|\mathbf{s}\|_1 \le 1\},$$

i.e., it is equal to the ℓ_1 unity norm-ball.

If $\mathbf{x}^* = 0$, the optimality condition in (2) implies that there exists $\mathbf{s}^* \in \mathbb{R}^n$ with $\|\mathbf{s}^*\|_1 \le 1$, such that

$$\mathbf{0} = \lambda\mathbf{s}^* - \mathbf{v} \Rightarrow \mathbf{v} = \lambda\mathbf{s}^*$$

As a result, this case holds only when

$$\|\mathbf{v}\|_1 = \sum_{k=1}^{n} |v_k| \le \lambda. \tag{4}$$

If $\mathbf{x}^* \neq 0$, focusing on (2), for the components with index $j \notin I$, the optimality condition simplifies to

$$0 = x_j^* - v_j \Rightarrow x_j^* = v_j, \tag{5}$$

i.e., proximal operator output component is equal to the corresponding input for $j \notin I$.

For the components with index $j \in I$, the optimality condition in (2) implies that there exists a β_j^* for which

$$x_j^* - v_j + \lambda sign(x_j)\beta_j^* = 0,$$

or equivalently,

$$x_j^* = v_j - \lambda sign(x_j^*)\beta_j^*. \tag{6}$$

Multiplying both sides with $sign(x_j^*)$ yields:

$$|x_j^*| = sign(x_j^*)v_j - \lambda\beta_j^*.$$

Now, we observe that $sign(x_k^*)$ must be equal to $sign(v_k)$. Because, if x_k^* had opposite sign, that would cause extra cost on the objective function compared to the equal signed version due to the quadratic term $\|\mathbf{x} - \mathbf{v}\|_2^2$. In other words, there always exists an equal signed version of x_k^* which yields same cost on $\|\mathbf{x}\|_\infty$ and less cost on $\|\mathbf{x} - \mathbf{v}\|_2^2$ relative to the opposite signed version. Therefore, replacing $sign(x_k^*)$ with $sign(v_k)$, we obtain

$$|x_j^*| = |v_j| - \lambda\beta_j^*. \tag{7}$$

Based on the definition of the index set I, $\forall k, l \in I; |x_k^*| = |x_l^*|$, and therefore, we have:

$$|v_k| - \lambda\beta_k^* = |v_l| - \lambda\beta_l^*$$

which leads to

$$\beta_k^* = \beta_l^* + \frac{|v_k| - |v_l|}{\lambda}.$$

Summing the equations for $\forall l \in I$, and replacing $\sum_{l\in I} \beta_l^* = 1$, we obtain

$$L\beta_k^* = 1 + \frac{L|v_k| - \sum_{l\in I}|v_l|}{\lambda},$$

where $L = card(I)$. Dividing by L, we get

$$\beta_k^* = \frac{|v_k| - \overline{v}}{\lambda} + \frac{1}{L},$$

where $\overline{v} = \frac{1}{L}\sum_{l\in I}|v_l|$. Inserting this expression in (6), we obtain

$$x_k^* = sign(v_k)(\overline{v} - \frac{\lambda}{L}). \tag{8}$$

The relation $sign(x_k^*) = sign(v_k)$ requires

$$\overline{v} > \frac{\lambda}{L}. \tag{9}$$

Equation (8) implies all the components corresponding to the index set I, are set to the same magnitude level $\mu = \overline{v} - \frac{\lambda}{L}$.

Moreover, $\forall j \notin I, k \in I$, we can write

$$|x_j^*| < |x_k^*|, \tag{10}$$

$$|v_j| < (\overline{v} - \frac{\lambda}{L}) \leq |v_k|, \tag{11}$$

where the inequality on the right is obtained from (7). Therefore, based on (11), we can conclude that index set I corresponds to the indexes of the L largest magnitude elements of $\mathbf{v}$. In effect, this fact together with equalities (5) and (8) imply that the L largest magnitude elements of $\mathbf{v}$ are clipped to the same level

$$\mu = \overline{v} - \frac{\lambda}{L}. \tag{12}$$

by the clipping function in (3) (and shown in Fig. 1). Therefore, $x_k = c_\mu(v_k)$ (or $\mathbf{x} = C_\mu(\mathbf{v})$).

As a result, what remains is to identify L, i.e., the cardinality of I, which also identifies the clipping level μ. For this purpose, we first define the vector $\mathbf{v}^{(s)} \in \mathbb{R}^n$ which is obtained by taking the absolute values of the elements of $\mathbf{v}$, and sorting them in a descending manner, i.e., $v_1^{(s)} \geq v_2^{(s)} \geq \ldots v_n^{(s)}$. In terms of $\mathbf{v}^{(s)}$, the inequalities in (11) can be reorganized as

$$\sum_{k=1}^{L-1} v_k^{(s)} - (L-1)v_L^{(s)} - \lambda \leq 0 \quad \text{if} \quad 2 \leq L \leq n \tag{13}$$

$$\sum_{k=1}^{L} v_k^{(s)} - Lv_{L+1}^{(s)} - \lambda > 0 \quad \text{if} \quad 1 \leq L \leq n-1. \tag{14}$$

We consider the following cases for L:

- If $L = 1$, then (14) implies

$$v_1^{(s)} - v_2^{(s)} - \lambda > 0. \tag{15}$$

- If $L \in \{2, \ldots, n-1\}$, we can rewrite the conditions in (13-14) as

$$\begin{bmatrix} 1 & .. & 1 & -L+1 & 0 & 0 & .. & 0 \\ 1 & .. & 1 & 1 & -L & 0 & .. & 0 \end{bmatrix} \mathbf{v}^{(s)} - \lambda\mathbf{1} \begin{matrix} \leq \\ > \end{matrix} 0. \tag{16}$$

- If $L = n$, the inequality in (13) implies

$$\sum_{k=1}^{n-1} v_k^{(s)} - (n-1)v_n^{(s)} - \lambda \leq 0, \tag{17}$$

and the inequality in (9) implies

$$\sum_{k=1}^{n} v_k^{(s)} - \lambda > 0. \tag{18}$$

In order to establish the choice of L, based on the inequalities in (15-18), we define

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & -2 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 1 & -(n-1) \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

and $\mathbf{z} = \mathbf{A}\mathbf{v}^{(s)} - \lambda\mathbf{1}$.

Regarding $\mathbf{z} \in \mathbb{R}^n$, we make the following useful observation: $\mathbf{z}$ has non-decreasing elements (with increasing index). This can be demonstrated by checking

$$z_{i+1} - z_i = \begin{cases} (i+1)(v_i^{(s)} - v_{i+1}^{(s)}), & i \leq n-1, \\ n v_n^{(s)} & i = n, \end{cases} \tag{19}$$

which is non-negative as $\mathbf{v}^{(s)}$ is non-increasing with non-negative elements. Based on the definition of $\mathbf{z}$ and its monotonic property, we can list the following three mutually exclusive cases to identify L:

i. The sign switching ($z_{k-1} \leq 0$ and $z_k > 0$ for $k \in \{2, \dots, n\}$): Based on the inequalities (16-18), there is a sign switching (from non-positive to positive) $\mathbf{z}$ when L is in $\{2, \dots, n\}$. In this case identifies L as the smallest index of $\mathbf{z}$ with positive value.

ii. All elements are positive $\mathbf{z} > \mathbf{0}$: This case identifies L as equal to 1. If $L = 1$, due to (15), $z_1 > 0$ and therefore $\mathbf{z} > \mathbf{0}$ due to its non-decreasing property.

iii. All elements are non-positive ($\mathbf{z} \leq \mathbf{0}$): This case implies $z_n \leq 0$, and therefore, $\|\mathbf{v}\| \leq \lambda$. By the inequality (4), this identifies the case $\mathbf{x}^* = 0$. Note that this case corresponds to clipping all elements (and therefore, $L = n$) with level $\mu = 0$.

We note that the clipping level formula in (12), which is derived for $\mathbf{x}^* \neq 0$, can be modified as

$$\mu = \{\overline{v} - \frac{\lambda}{L}\}^+ = \frac{1}{L}\{\mathbf{1}_L{}^T\mathbf{v}^{(s)} - \lambda\}^+, \tag{20}$$

so that the case iii, i.e., $\mathbf{x}^* = 0$, can be covered with this formula as the argument of the rectified linear mapping would be negative which would correspond to 0 clipping level ■

The algorithm implementation for the proximal operator consists of three main steps: 1.) Sorting $\mathbf{v}$ to obtain $\mathbf{v}^{(s)}$, 2. Identifying L and μ, and 3. Clipping operation with the computed μ. The main computational complexity is due to sorting in the first step (which requires $O(nlog(n))$ operations) and the second step, where the computation of $\mathbf{z}$ can be greatly simplified (to $O(n)$ operations) by using the recursion in (19).

As a final note, the above proximal operator derived for real vectors can be easily extended to the complex vectors. Skipping the derivation due to length constraints, only difference is essentially that the clipping expression (8) is replaced with

$$x_k^* = e^{j\theta(v_k)}(\overline{v} - \frac{\lambda}{L}).$$

where $v_k = |v_k|e^{j\theta(v_k)}$, and therefore, the complex version of the elementwise clipping function is equal to

$$c_\mu(x_k) = \begin{cases} v_k & |v| \leq \mu, \\ \mu e^{j\theta(v_k)} & \text{otherwise.} \end{cases}$$

4. NUMERICAL EXAMPLE

As an example to illustrate the benefit of ℓ_∞-norm operator, we consider the ℓ_∞-norm based blind equalization problem [1]. In the corresponding setting, we assume $4QAM$ signals are sent through an LTI channel whose output is corrupted by Gaussian noise. In the blind equalization, the purpose is to invert the channel adaptively, using only measurements at the receiver, without any training informations. In the convex blind equalization approach of [1], the ℓ_∞-norm of the equalizer output is minimized while one of the taps of the equalizer is fixed as constant. The corresponding optimization problem can be formulated as the minimization of $\|\mathbf{Y}\mathbf{w}+\mathbf{y}\|_\infty$, where $\mathbf{w}$ is the non-fixed components of the equalizer filter, $\mathbf{Y}$ and $\mathbf{y}$ are the is the sub-matrices of the convolution matrix containing measurements corresponding to free and fixed parts of the equalizer impulse response. As an example we consider the 4-tap complex channel $\mathbf{h} = [-1.0493 + 0.2305i \; 1.4129 - 1.4497i \; - 0.2540 + 0.2021i \; 0.5302 - 0.7732i]^T$ [16]. In Fig. 2, we compare average ISI values (the ratio of the residual ISI energy to signal energy) per iteration for Subgradient based [4] and ADMM[10, 17] algorithm using the ℓ_∞-norm proximal operator, which shows that the proximal operator clearly improves the convergence behavior in terms of both target level and iteration count.

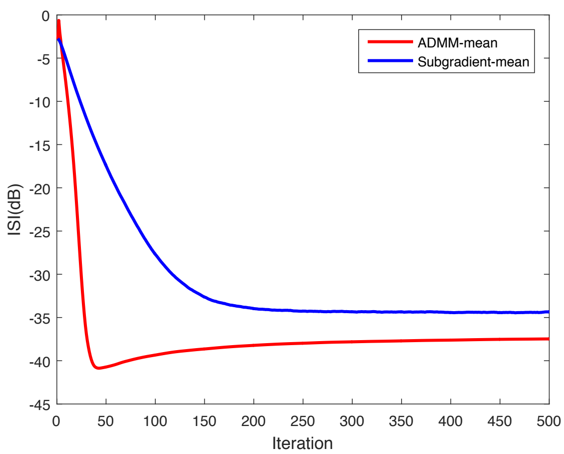

Fig. 2. ISI(dB) ADMM vs Subgradient

5. REFERENCES

[1] S. Vembu, S. Verdu, R. Kennedy, and W. Sethares, "Convex cost functions in blind equalization," *IEEE Trans. on Signal Processing*, vol. 42, pp. 1952–1960, August 1994.

[2] Z. Ding and Z. Luo, "A fast linear programming algorithm for blind equalization," *IEEE TCOM.*, vol. 48, pp. 1432–1436, September 2000.

[3] Zhi-Quan Luo, Mei Meng, Kon Max Wong, and Jian-Kang Zhang, "A fractionally spaced blind equalizer based on linear programming," *IEEE Trans. on Signal Processing*, vol. 50, pp. 1650–1660, July 2002.

[4] Alper T. Erdogan and Can Kizilkale, "Fast and low complexity blind equalization via subgradient projections," *IEEE Trans. on Signal Processing*, vol. 53, pp. 2513–2524, July 2005.

[5] Alper T. Erdogan, "A simple geometric blind source separation method for bounded magnitude sources," *IEEE Trans. on Signal Processing*, vol. 54, pp. 438–449, February 2006.

[6] Alper T. Erdogan, "A class of bounded component analysis algorithms for the separation of both independent and dependent sources," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5730–5743, 2013.

[7] Baki B Yilmaz and Alper T Erdogan, "Compressed training adaptive equalization," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4920–4924.

[8] B. B. Yilmaz and A. T. Erdogan, "Compressed training adaptive equalization: Algorithms and analysis," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3907–3921, Sept 2017.

[9] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[10] Neal Parikh, Stephen Boyd, et al., "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[11] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, New York, NY, USA, 2008, ICML '08, pp. 272–279, ACM.

[12] Shai Shalev-Shwartz and Yoram Singer, "Efficient learning of label ranking by soft projections onto polyhedra," *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1567–1599, 2006.

[13] Tom Goldstein, Min Li, Xiaoming Yuan, Ernie Esser, and Richard Baraniuk, "Adaptive primal-dual hybrid gradient methods for saddle-point problems," *arXiv preprint arXiv:1305.0546*, 2013.

[14] Ender Erkaya and Alper T. Erdogan, "Proximal operator for the k-peak norm," *In preperation*.

[15] Tom Goldstein, Christoph Studer, and Richard G. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," *CoRR*, vol. abs/1411.3406, 2014.

[16] John G. Proakis, *Digital Communications*, McGraw Hill, 2000.

[17] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.