Comments on Efficient Singular Value Thresholding Computation

Zhengyuan Zhou∗ Yi Ma†

Abstract

We discuss how to evaluate the proximal operator of a convex and increasing function of a nuclear norm, which forms the key computational step in several first-order optimization algorithms such as (accelerated) proximal gradient descent and ADMM. Various special cases of the problem arise in low-rank matrix completion, dropout training in deep learning and high-order low-rank tensor recovery, although they have all been solved on a case-by-case basis. We provide an unified and efficiently computable procedure for solving this problem.

1 Problem, Notation and Background

Proximal gradient descent (and its accelerated variant) (Beck and Teboulle (2009); Combettes and Pesquet (2011); Ma (2012); Parikh and Boyd (2014)) provide an efficient way (with $O(\frac{1}{T})$ and $O(\frac{1}{T^2})$ convergence rates, respectively) to solve structured convex non-smooth optimization problems of the following form, which arise frequently in machine learning and structured signal recovery settings:

$$
\min_x F(x) = g(x) + h(x),
$$

where $g(x)$ is convex and smooth (i.e. continuously differentiable with bounded gradients) and $h(x)$ is convex but non-smooth. The presence of $h$ often results from non-smooth but convex regularizers such as $l_1$-norm or nuclear norm, as two prominent examples. The computational bottleneck in each iteration of (accelerated) proximal gradient is evaluating the proximal operator (at a point $w$):

$$
\text{prox}_h[w] = \arg \min_x \{h(x) + \frac{1}{2}\|x - w\|^2\},
$$

which admits a unique solution due to strong convexity. Computationally, efficient (accelerated) proximal gradient is possible when and only when $\text{prox}_h[w]$ can be computed efficiently. Note that in addition to (accelerated) proximal gradient, evaluating proximal operators is also the key and computationally demanding step in methods such as ADMM (Luo (2012); Yang and Yuan (2013); Boyd et al. (2011)) for large-scale equality constrained optimization problems.

In this note, we focus on a class of non-smooth convex functions $h : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$, where $h(X) = f(\|X\|_\ast)$ for some convex and increasing $f : \mathbb{R} \to \mathbb{R}$, with $\|X\|_\ast$ being the nuclear norm of $X$. Thus, our main question is, how to efficiently compute the optimal solution $X^*$, where:

$$
X^* = \arg \min \{\tau f(\|X\|_\ast) + \frac{1}{2}\|X - Y\|^2_F\}. \quad (1.1)
$$

* Stern School of Business, New York University. Email: zzhou@stern.nyu.edu.
† UC Berkeley EECS. Email: yima@eecs.berkeley.edu
It turns out a few special cases of this problem have played important roles in machine learning and signal processing. For instance, when \( f(x) = x \), this problem occurs in low-rank matrix completion and recovery problems (Cai et al. (2010)); when \( f(x) = x^2 \), this problem occurs in drop-out training in deep learning (Cavazza et al. (2018)); when \( f(x) = e^x \), this problem occurs in high-order low-rank tensor recovery (Zhang et al. (2014)). Further, analytical and/or efficiently computable solutions have been derived in these special cases. For instance, consider a matrix \( Y \in \mathbb{R}^{n_1 \times n_2} \) of rank \( r \), whose singular value decomposition (SVD) is:

\[
Y = U\Sigma V^*, \Sigma = \text{diag}(\{\sigma_i^Y\}_{1 \leq i \leq r}), \sigma_1^Y \geq \sigma_2^Y \geq \cdots \geq \sigma_r^Y > 0,
\]

where \( U \) and \( V \) are \( n_1 \times r \) and \( n_2 \times r \) matrices respectively with orthonormal columns. For each \( \tau > 0 \), the soft thresholding shrinkage operator \( D_\tau \) is defined to be:

\[
D_\tau(Y) = U D_\tau(\Sigma) V^*, \quad D_\tau(\Sigma) = \text{diag}((\sigma_i^Y - \tau)_+), t_+ = \max(0, t).
\]

Cai et al. (2010) shows that soft thresholding provides an analytical solution when \( f(x) = x \).

**Lemma 1.** [Cai et al. (2010)] Given a matrix \( Y \in \mathbb{R}^{n_1 \times n_2} \) and a \( \tau \geq 0 \), consider the function \( h(X) = \tau \|X\|_{\infty} + \frac{1}{2}\|X - Y\|_F^2 \), \( X \in \mathbb{R}^{n_1 \times n_2} \). We have \( \arg \min_X h(X) = D_\tau(Y) \).

However, although the proximal mappings of these different cases have been solved separately, there is a lack of an unified and efficiently computable scheme to solve the above problem for a general \( f \) (one should not expect an analytical solution exists for a general \( f \)). It turns out that soft singular value thresholding still works and the threshold can be computed efficiently by a binary search on a system of 1-dimensional equations that depend on the input data matrix \( Y \) and \( f(\cdot) \).

### 2 Main Results

We augment the list of non-increasing singular values \( \{\sigma_i^Y\}_{1 \leq i \leq r} \) of \( Y \) with \( \sigma_r^{Y+1} = -\infty \).

**Lemma 2.** Let \( g : \mathbb{R} \rightarrow \mathbb{R} \) be such that \( g(x) \geq 0 \), \( g \) is increasing and \( g(0) \leq 1 \). Suppose the rank of \( Y \) is \( r \) and \( \tau < \sigma_1^Y \). There exists a unique integer \( j \), with \( 1 \leq j \leq r \), such that the solution \( t_j \) to the following equation

\[
g(\sum_{i=1}^{j} \sigma_i^Y - j t_j) = \frac{t_j}{\tau}
\]

satisfies the constraint

\[
\sigma_j^{Y+1} \leq t_j < \sigma_j^Y.
\]

**Proof.** We first show that if at least one such \( j \) exists, then such a \( j \) (and hence \( t_j \)) is unique. Consider the set \( J = \{ j \mid t_j \) satisfies (2.1) and (2.2)\}. Assume \( J \neq \emptyset \), let \( j^* \) be the smallest element in \( J \). Now we argue that no \( j^* + k \), \( 1 \leq k \leq r - j^* \), can be in \( J \). Consider any \( k \) with \( 1 \leq k \leq r - j^* \). Suppose for contradiction \( j^* + k \in J \). That is:

\[
g(\sum_{i=1}^{j^*+k} \sigma_i^Y - (j^* + k) t_{j^*+k}) = \frac{t_{j^*+k}}{\tau},
\]

\[
\sigma_{j^*+k+1}^{Y} \leq t_{j^*+k} < \sigma_{j^*+k}^{Y}.
\]

2
Expanding on the right side of (2.3), we have
\[
g(\sum_{i=1}^{j*+k} \sigma_{j*} - (j* + k)t_{j*+k}) \geq g(\sum_{i=1}^{j*} \sigma_{j*} + k\sigma_{j*+k} - (j* + k)t_{j*+k})
\]
\[
= g(\sum_{i=1}^{j*} \sigma_{j*} - j*t_{j*+k} + k(\sigma_{j*+k} - t_{j*+k})) > g(\sum_{i=1}^{j*} \sigma_{j*} - j*t_{j*+k})
\]
\[
> g(\sum_{i=1}^{j*} \sigma_{j*} - j*t_{j*}) = \frac{t_{j*}}{\tau},
\]
where the first inequality follows from the non-increasing values of the singular values, the second inequality follows from the assumption in (2.4) and that \(g\) is increasing, the last inequality follows from \(t_{j*} \geq \sigma_{j*+1}^2 \geq \sigma_{j*+k}^2 > t_{j*+k}\) and the last equality follows from the definition of \(t_{j*}\).

Hence, it follows that
\[
\frac{t_{j*+k}}{\tau} = g(\sum_{i=1}^{j*+k} \sigma_{j*} - (j* + k)t_{j*+k}) > \frac{t_{j*}}{\tau},
\]
leading to \(t_{j*+k} > t_{j*}\), hence a contradiction.

Next, we prove that \(J\) is indeed not empty.

First, we note that by the property of \(g\), a unique solution \(t_{j} > 0\) exists for \(g(\sum_{i=1}^{j} \sigma_{j} - y_{j}) = \frac{y_{j}}{\tau}\), for each \(j\) satisfying \(1 \leq j \leq r\). We denote by \(t_{j}\) the unique solution corresponding to each \(j\).

Hence, it suffices to show at least one \(t_{j}\) satisfies \(\sigma_{j+1}^{2} \leq t_{j} < \sigma_{j}^{2}\).

Again by monotonicity of \(g\) and \(g(0) \leq 1\), it is easily seen that \(\tau < \sigma_{1}^{2}\) implies that \(t_{1} < \sigma_{1}^{2}\). Now suppose it also holds that \(\sigma_{1}^{2} \leq t_{1}\), then we are done. Otherwise, we have \(t_{1} < \sigma_{1}^{2}\). Under this assumption, we claim that \(t_{1} < t_{2}\) and \(t_{2} < \sigma_{2}^{2}\). To prove \(t_{1} < t_{2}\), assume for the sake of contradiction that \(t_{1} \geq t_{2}\), leading to:
\[
\frac{t_{1}}{\tau} = g(\sigma_{1}^{2} - t_{1}) < g(\sigma_{1}^{2} + \sigma_{2}^{2} - 2t_{2})
\]
\[
\leq g(\sigma_{1}^{2} + \sigma_{2}^{2} - 2t_{2}) = \frac{t_{2}}{\tau},
\]
where the first inequality follows from \(t_{1} < \sigma_{1}^{2}\). Hence we reach a contradiction, establishing that \(t_{1} < t_{2}\).

The desired inequality \(t_{2} < \sigma_{1}^{2}\) then follows since
\[
g(\sigma_{1}^{2} - t_{1}) = \frac{t_{1}}{\tau} < \frac{t_{2}}{\tau} < g(\sigma_{1}^{2} + \sigma_{2}^{2} - t_{1} - t_{2})
\]
implies \(\sigma_{1}^{2} - t_{1} < \sigma_{1}^{2} + \sigma_{2}^{2} - t_{1} - t_{2}\) by monotonicity of \(g\), hence yielding \(t_{2} < \sigma_{2}^{2}\).

If \(t_{2} \geq \sigma_{2}^{2}\), then the claim is established. If not, we can repeat this process inductively. More formally, suppose we have just finished the \(j\)-th iteration (note that the induction basis \(j = 1\) is verified above) and we have \(t_{j} < \sigma_{j}^{2}\). If it also holds that \(t_{j} \geq \sigma_{j+1}^{2}\), then the claim follows. If not,
then we show \( t_{j+1} > t_j \) and \( t_{j+1} < \sigma_Y^{j+1} \). First, assume on the contrary, \( t_{j+1} \leq t_j \)

\[
\frac{t_j}{r} = g(\sum_{i=1}^{j} \sigma_Y^{-} - j t_j) < g(\sum_{i=1}^{j+1} \sigma_Y^{+} - (j + 1) t_{j+1}) = \frac{t_{j+1}}{\tau}
\]

where the first inequality follows from \( t_j < \sigma_Y^{j} \). Hence we reach a contradiction, establishing that \( t_j < t_{j+1} \).

Next, we note that \( t_{j+1} < \sigma_Y^{j+1} \) follows since

\[
g(\sum_{i=1}^{j} \sigma_Y^{+} - j t_j) = \frac{t_j}{\tau} < \frac{t_{j+1}}{\tau} < g(\sum_{i=1}^{j+1} \sigma_Y^{+} - j t_j - t_{j+1}),
\]

(where the last inequality follows due to \( j t_j < j t_{j+1} \)) implying \( \sum_{i=1}^{j} \sigma_Y^{+} - j t_j < \sum_{i=1}^{j+1} \sigma_Y^{+} - j t_j - t_{j+1} \), which is equivalent to \( t_{j+1} < \sigma_Y^{j+1} \).

Thus, we have a strictly increasing sequence \( \{t_j\} \) with \( t_j < \sigma_Y^{j} \). If it holds that \( \sigma_Y^{j+1} \leq t_j < \sigma_Y^{j} \) at some iteration \( j \), then such a \( j \) certifies that \( J \) is not empty. If \( \sigma_Y^{j+1} \leq t_j < \sigma_Y^{j} \), never holds for \( j \) up to \( r - 1 \), then it must hold for \( j = r \), since \(-\infty = \sigma_Y^{r+1} \leq t_r < \sigma_Y^{r} \), also certifying that \( J \) is not empty.

**Remark 1.** In addition to asserting the unique existence of such a \( j^{*} \), the proof suggests a natural binary search algorithm to find such a \( j^{*} \) and the corresponding \( t_{j^{*}} \). The algorithm is given in Algorithm 1. Note that the step “Compute \( t_{j^{*}} \) can be easily done very efficiently by numerically solving \( g(\sum_{i=1}^{j} \sigma_Y^{+} - j t_j) = \frac{t_j}{\tau} \), even though there may not be any analytical solution.

**Lemma 3.** Algorithm 1 correctly computes the unique \( j \) and \( t_j \) guaranteed by Lemma 2.

**Proof.** From the first part of proof for Lemma 2, we know that if \( j^{*} \) is the unique \( j \) guaranteed by Lemma 2, then for all \( k > j^{*} \), we have \( t_k \geq \sigma_Y^{k} \). Thus, if \( t_M < \sigma_Y^{M} \), then we know that \( j^{*} \) cannot be less than \( M \). That is, \( j^{*} \) must be in the second half of the unsearched space. Conversely, if we hypothetically do a sequential search, then it follows immediately from the second part of proof of Lemma 2 that before \( j \) reaches \( j^{*} \), \( t_M < \sigma_Y^{M} \) must hold. This establishes that if in the while loop we encounter \( t_M \geq \sigma_Y^{M} \), then it must be the case that \( j^{*} \leq M \). That is, \( j^{*} \) must lie in the first part of the unsearched space. It then follows that \( j^{*} \) always lies between \( L \) and \( R \), establishing that while loop will eventually halt, returning \( t_{j^{*}} \) and \( j^{*} \).

**Definition 1.** Given \( \tau > 0 \), the generalized singular value thresholding operator \( \mathcal{H}_{\tau} \) is defined to be

\[
\mathcal{H}_{\tau}(Y) = UD_{t^{*}}(\Sigma)V^{*}, \quad Y = U\Sigma V^{*} \in \mathbb{R}^{n_1 \times n_2},
\]

where \( t^{*} \) is the threshold computed by Algorithm 1.

Lemma 2 guarantees that \( \mathcal{H}_{\tau} \) is well-defined and Algorithm 1 guarantees that \( \mathcal{H}_{\tau} \) is efficiently computable. Having defined \( \mathcal{H}_{\tau} \), the main result is:

**Theorem 1.** Let \( f : \mathbb{R} \rightarrow \mathbb{R} \) be any convex, increasing, differentiable function, with an increasing derivative satisfying \( f'(0) \leq 1 \). Given a \( \tau > 0 \) and a \( Y \in \mathbb{R}^{n_1 \times n_2} \), we have:

\[
\mathcal{H}_{\tau}(Y) = \arg \min_X \{ \tau f(\|X\|_1) + \frac{1}{2} \|X - Y\|_F^2 \}.
\]
Algorithm 1 Generalized Singular Value Threshold Computation

\begin{algorithm}
\begin{algorithmic}
\Require $Y$, $\tau$, $\{\sigma^i_Y\}_{1 \leq i \leq r}$
\Ensure $j^*$ and $t_j^*$
\If{$\tau \geq \sigma^1_Y$, \Return 0 and $\sigma^1_Y$}
\EndIf
\Initialize $L = 1$, $R = r$,
\While{$L < R$} 
\State $M = \lceil \frac{L + R}{2} \rceil$, and compute the solution $t_M$ to the equation $g(\sum_{i=1}^M \sigma^i_Y - Mt_M) = \frac{t_M}{\tau}$
\If{$t_M < \sigma^M_Y$ 
\Then
\If{$\sigma^M_Y + 1 \leq t_M$, \Return $M$ and $t_M$}
\Else $L = M$, \Continue
\EndIf
\Else $R = M$, \Continue
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}

Proof. To prove this theorem, we build on the techniques introduced in Cai et al. (2010).

The function $h(X) = \tau f(||X||_*) + \frac{1}{2} ||X - Y||_F^2$ is strictly convex, since it is the sum of a convex function and a strictly convex function. As a result, the minimizer $\hat{X}$ to $h(X)$ is unique and it suffices to show that $\mathcal{H}_+(Y)$ is one minimizer.

Per the definition of a subgradient, $S$ is a subgradient of a convex function $f$ at $X_0$ if $f(X) \geq f(X_0) + \langle S, X - X_0 \rangle$. $\partial f(X_0)$ is commonly used to denote the set of subgradients of $f$ at $X_0$. Recall that the set $\partial||X||_*$, of subgradients of the nuclear norm function at $X_0$ is: $\partial||X||_* = \{ U_X \Sigma_X V_X^* + W | W \in \mathbb{R}^{n_1 \times n_2}, U_X W = 0, W V_X = 0, ||W||_2 \leq 1 \}$, where the SVD of $X_0$ is $U_x \Sigma_x V_x^*$ and $||W||_2$ is the top singular value of $W$.

First, it is easy to check that $f'(||X||_*)(U_X V_X^* + W) \in \partial(f(||X||_*))$ for $W \in \mathbb{R}^{n_1 \times n_2}, U_X W = 0, W V_X = 0, ||W||_2 \leq 1$, by the composition rule for the subgradient. Hence, $\tau g(||\hat{X}||_*)(U_X V_X^* + W) + \hat{X} - Y$ is a subgradient for $h$ at $\hat{X}$, for $W$ satisfying $U_X W = 0, W V_X = 0, ||W||_2 \leq 1$, where $g = f'$ and $g$ satisfies the assumption given in Lemma 2. Moreover, if there exists such a $W$ and it holds that $0 = \tau g(||\hat{X}||_*)(U_X V_X^* + W) + \hat{X} - Y$, or equivalently that

$$Y - \hat{X} = \tau g(||\hat{X}||_*)(U_X V_X^* + W), \quad (2.5)$$

then $\hat{X}$ is a minimizer (hence the unique minimizer) to $h(X)$.

We now establish that, with $\hat{X} = \mathcal{H}_+(Y)$, Eq. (2.5) does hold with $W$ satisfying the given constraints. First, we consider the case that $\tau < \sigma^1_Y$.

By Lemma 2, since $t_j^*$ satisfies the equation $\frac{t_j^*}{\tau} = g(\sum_{i=1}^{j^*} \sigma^i_Y - j^* t_j^*)$, we have $t_j^* = \tau g(\sum_{i=1}^{j^*} (\sigma^i_Y - t_j^*))$.

Since $\sigma^{j^*+1}_Y \leq t_j^*$, $\sigma^{j^*}_Y$’s last $r - j^*$ singular values ($\sigma^{k}_Y$ with $k \geq j^* + 1$) have been set to 0, leading to that $||\hat{X}||_* = \sum_{i=1}^{j^*} (\sigma^i_Y - t_j^*)$. Therefore, $t_j^* = \tau g(||\hat{X}||_*)$.

Next, we partition $Y$ as follows:

$$Y = U_a \Sigma_a V_a^* + U_b \Sigma_b V_b^*,$$

where $U_a$’s columns and $V_a$’s columns are the first $j^*$ left and right singular vectors respectively, associated with the first $j^*$ singular values (i.e. singular values larger than $t_j^*$), while $U_b$’s columns
and $V_b$’s columns are the remaining $r - j^*$ left and right singular vectors respectively, associated with the remaining $r - j^*$ singular values (i.e. singular values less than or equal to $t_{j^*}$).

Under this partition, it is easily seen that $\hat{X} = U_a (\Sigma_a - t_{j^*} I) V_a^*$. We then have

$$Y - \hat{X} = t_{j^*} U_a V_a^* + U_b \Sigma_b V_b^*$$

$$= \tau g(\|\hat{X}\|_*) U_a V_a^* + U_b \Sigma_b V_b^*$$

$$= \tau g(\|\hat{X}\|_*) (U_a V_a^* + \frac{1}{t_{j^*}} U_b \Sigma_b V_b^*).$$

Choose $W = \frac{1}{t_{j^*}} U_b \Sigma_b V_b^*$. By construction, $U_a^* U_b = 0$, $V_b^* V_a = 0$, hence we have $U_a^* W = 0, W V_{\hat{X}} = 0$. In addition, since $t_{j^*} \geq \sigma_{j^*+1}$, we have $\|W\|_2 = \frac{\sigma_{j^*+1}}{t_{j^*}} \leq 1$. Hence $W$ thus chosen satisfies the constraints, hence establishing the claim.

Now, if $\tau \geq \sigma_Y^1$, then $t_{j^*} = \sigma_Y^1$ and $j^* = 1$ are returned by Algorithm 1. It follows immediately that in this case $\hat{X} = 0$. Choosing $W = \tau^{-1} Y$, it is easily seen that $W$ satisfies the constraints. Verification of Eq. (2.5) is instant when $\hat{X} = 0$ and $W = \tau^{-1} Y$.

References

Beck, A. and Teboulle, M. (2009). Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, pages 42–88.

Boyd, S. P., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Cai, J., Candès, E., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Cavazza, J., Morerio, P., Haefele, B., Lane, C., Murino, V., and Vidal, R. (2018). Dropout as a low-rank regularizer for matrix factorization. In *International Conference on Artificial Intelligence and Statistics*, pages 435–444. PMLR.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer.

Luo, Z. (2012). On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*.

Ma, S. (2012). Alternating proximal gradient method for convex minimization. *Preprint of Optimization Online*.

Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239.

Yang, J. and Yuan, X. (2013). Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329.

Zhang, X., Zhou, Z., Wang, D., and Ma, Y. (2014). Hybrid singular value thresholding for tensor completion. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. Citeseer.