Efficient Learning with a Family of Nonconvex Regularizers by Redistributing Nonconvexity

Quanming Yao   QYAOAA@CSE.UST.HK
James T. Kwok   JAMESK@CSE.UST.HK
Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

Abstract

The use of convex regularizers allow for easy optimization, though they often produce biased estimation and inferior prediction performance. Recently, nonconvex regularizers have attracted a lot of attention and outperformed convex ones. However, the resultant optimization problem is much harder. In this paper, for a large class of nonconvex regularizers, we propose to move the nonconvexity from the regularizer to the loss. The nonconvex regularizer is then transformed to a familiar convex regularizer, while the resultant loss function can still be guaranteed to be smooth. Learning with the convexified regularizer can be performed by existing efficient algorithms originally designed for convex regularizers (such as the standard proximal algorithm and Frank-Wolfe algorithm). Moreover, it can be shown that critical points of the transformed problem are also critical points of the original problem. Extensive experiments on a number of nonconvex regularization problems show that the proposed procedure is much faster than the state-of-the-art nonconvex solvers.

1. Introduction

In many machine learning models, the associated optimization problems are of the form

$$\min_{x \in \mathbb{R}^d} F(x) \equiv f(x) + g(x), \qquad (1)$$

where x is the model parameter, f is the loss and g is the regularizer. Obviously, the choice of regularizers is important and application-specific. For example, sparsity-inducing regularizers are commonly used on high-dimensional data (Jacob et al., 2009); while low-rank regularizers are used in the learning of matrices (Candès & Recht, 2009) and tensors (Liu et al., 2013).

Traditionally, both the loss and regularizer are convex, leading to the popularity of convex optimization tools in machine learning (Boyd & Vandenberghe, 2004). Prominent examples include the proximal algorithm (Parikh & Boyd, 2013) and, more recently, the Frank-Wolfe (FW) algorithm (Jaggi, 2013). Many of them are efficient, scalable, and have sound convergence properties.

However, the use of convex regularizers often produces biased estimation, and produces solutions that are not as sparse and accurate as desired (Zhang, 2010b). To alleviate this problem, a number of nonconvex regularizers have been recently proposed (Geman & Yang, 1995; Fan & Li, 2001; Candès et al., 2008; Zhang, 2010a; Trzasko & Manduca, 2009). They are all (i) nonsmooth at zero, which encourages a sparse solution; and (ii) concave, which places a smaller penalty than the ℓ_1 regularizer on features with large magnitudes. Empirically, they usually outperform convex regularizers.

However, the resulting nonconvex problem is much harder to optimize. The concave-convex procedure (Yuille & Rangarajan, 2002) is a general technique for nonconvex optimization. However, at each iteration, the subproblem can be as expensive as the original problem, and are thus slow in practice (Gong et al., 2013; Zhong & Kwok, 2014). Recently, proximal algorithms have also been extended for nonconvex regularization (Gong et al., 2013; Li & Lin, 2015). However, efficient computation of the underlying proximal operator is only possible for simple nonconvex regularizers. When the regularizer is complicated, such as the nonconvex versions of the fused lasso or overlapping group lasso regularizers, the proximal step has to be solved numerically and is again expensive. Another approach is by using the proximal average (Zhong & Kwok, 2014), which computes and averages the proximal step of each underlying regularizer separately. However, because of its approximate proximal step, convergence is usually slower than typical applications of the proximal algorithm.

In this paper, we propose to handle nonconvex regularizers by reusing the abundant repository of efficient convex algorithms originally designed for convex regularizers. The key is to shift the nonconvexity associated with the nonconvex regularizer to the loss function, and transform the nonconvex regularizer to a familiar convex regularizer. It will be shown that every critical point of the transformed problem is also a critical point of the original problem.

To illustrate the practical usefulness of this convexification scheme, we show how it can be used with the proximal algorithm for nonconvex structured sparse learning, and with the FW algorithm for matrix learning with nonconvex low-rank regularizers. As the FW algorithm has only been used on convex problems, we also propose a new FW variant that has guaranteed convergence to a critical point of the nonconvex problem.

Notation: We denote vectors and matrices by lowercase and uppercase boldface letters, respectively. For a vector $x \in \mathbb{R}^d$, $\|x\|_p = (\sum_{i=1}^{d} |x_i|^p)^{\frac{1}{p}}$ is its ℓ_p-norm. For a matrix $X \in \mathbb{R}^{m \times n}$ (where $m \leq n$ without loss of generality), its nuclear norm is $\|X\|_* = \sum_{i=1}^{m} \sigma_i(X)$, where $\sigma_i(X)$'s are the singular values of X, and its Frobenius norm is $\|X\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}^2}$. For two matrices X and Y, $\langle X, Y \rangle = \sum_{i,j} X_{ij} Y_{ij}$. For a smooth function f, $\nabla f(x)$ is its gradient at x. For a convex but nonsmooth f, $\partial f(x) = \{u : f(y) \geq f(x) + \langle u, y - x \rangle\}$ is its subdifferential at x, and $g \in \partial f(x)$ is a subgradient.

2. Shifting Nonconvexity from Regularizer to Loss

In recent years, a number of nonconvex regularizers have been proposed. Examples include the Geman penalty (GP) (Geman & Yang, 1995), log-sum penalty (LSP) (Candès et al., 2008), minimax concave penalty (MCP) (Zhang, 2010a), Laplace penalty (Trzasko & Manduca, 2009), and smoothly clipped absolute deviation (SCAD) penalty (Fan & Li, 2001). In general, learning with nonconvex regularizers is much more difficult than learning with convex regularizers. In this section, we show how to move the nonconvex component from the nonconvex regularizers to the loss function. Existing algorithms can then be reused to learn with the convexified regularizers.

First, we make the following standard assumptions on (1).

A1. F is bounded from below;

A2. f is L-Lipschitz smooth (i.e., $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$), but possibly nonconvex.

Let κ be a function that is concave, non-decreasing, ρ-Lipschitz smooth, and $\kappa(0) = 0$. Depending on the application, we consider g of the following forms.

C1. $g(x) = \sum_{i=1}^{K} \mu_i g_i(x)$, where $\mu_i \geq 0$,

$$g_i(x) = \kappa(\|A_i x\|_p), \qquad (2)$$

and A_i is a matrix. All the popular nonconvex regularizers in Table 1 satisfy this assumption. When κ is the identity function, $g(x)$ reduces to the convex regularizer $\sum_{i=1}^{K} \mu_i \|A_i x\|_p$. By using different A_i's, this becomes various structured sparsity regularizers such as group lasso (Jacob et al., 2009), fused lasso (Tibshirani et al., 2005), and graphical lasso (Jacob et al., 2009).

C2. $g(x) = \sum_{i=1}^{d} \kappa(|x_i|)$. When κ is the identity function, g becomes the lasso regularizer.

C3. $g(X) = \sum_{i=1}^{m} \kappa(\sigma_i(X))$, where X is a matrix. When κ is the identity function, g becomes the nuclear norm regularizer.

2.1. Key Idea

First, consider the form of g in **C1**. Rewrite each nonconvex g_i in (2) as

$$g_i(x) = \bar{g}_i(x) + \kappa_0 \|A_i x\|_p, \qquad (3)$$

where $\kappa_0 = \kappa'(0)$, and $\bar{g}_i(x) = \kappa(\|A_i x\|_p) - \kappa_0 \|A_i x\|_p$. Obviously, $\kappa_0 \|A_i x\|_p$ is convex but nonsmooth. The following shows that $\bar{g}_i$, though nonconvex, is concave and Lipschitz smooth.

Proposition 2.1. *For $p \in (1, +\infty)$, $\kappa(\|z\|_p) - \kappa_0 \|z\|_p$ is concave and $\bar{L}_i$-Lipschitz smooth.*

Corollary 2.2. *For $p \in (1, +\infty)$, (i) $\bar{g}_i$ is concave and $\bar{L}_i$-Lipschitz smooth; (ii) g can be decomposed as $g(x) = \bar{g}(x) + \breve{g}(x)$, where $\bar{g}(x) \equiv \sum_{i=1}^{K} \mu_i \bar{g}_i(x)$ is concave and Lipschitz smooth, while $\breve{g}(x) \equiv \kappa_0 \sum_{i=1}^{K} \mu_i \|A_i x\|_p$ is convex but possibly nonsmooth.*

Problem (1) can then be rewritten as[1]

$$\min_x \bar{f}(x) + \breve{g}(x), \qquad (4)$$

where $\bar{f}(x) \equiv f(x) + \bar{g}(x)$. Note that $\bar{f}$ (viewed as an augmented loss) is $\bar{L}$-Lipschitz smooth, where $\bar{L} = L + \sum_{i=1}^{K} \mu_i \bar{L}_i$; while $\breve{g}$ (viewed as a convexified regularizer) is convex but possibly nonsmooth. In other words, nonconvexity is shifted from the regularizer g to the loss f, while ensuring that the augmented loss is smooth. As will be demonstrated in the following sections, this allows

---

[1]In the sequel, a function with a bar on top (e.g., $\bar{f}$) denotes that it is smooth; whereas a function with breve (e.g., $\breve{g}$) denotes that it may be nonsmooth.

Table 1. Example nonconvex regularizers. Here, $\mu > 0$, and $\theta > 1$ for SCAD, and $\theta > 0$ for others.

	$\kappa(\alpha)$	$\kappa'(\alpha)$	κ_0	ρ
GP (Geman & Yang, 1995)	$\frac{\mu\alpha}{\theta+\alpha}$	$\frac{\mu\theta}{(\theta+\alpha)^2}$	$\frac{\mu}{\theta}$	$\frac{2\mu}{\theta^2}$
LSP (Candès et al., 2008)	$\mu\log(1+\frac{\alpha}{\theta})$	$\frac{\mu}{\theta+\alpha}$	$\frac{\mu}{\theta}$	$\frac{\mu}{\theta^2}$
MCP (Zhang, 2010a)	$\begin{cases} \mu\alpha - \frac{\alpha^2}{2\theta} & \alpha \leq \mu\theta \\ \frac{1}{2}\theta\mu^2 & \alpha > \mu\theta \end{cases}$	$\begin{cases} \mu - \frac{\alpha}{\theta} & \alpha \leq \mu\theta \\ 0 & \alpha > \mu\theta \end{cases}$	μ	$\frac{1}{\theta}$
Laplace (Trzasko & Manduca, 2009)	$\mu(1 - \exp(-\frac{\alpha}{\theta}))$	$\frac{\mu}{\theta}\exp\left(-\frac{\alpha}{\theta}\right)$	$\frac{\mu}{\theta}$	$\frac{\mu}{\theta^2}$
SCAD (Fan & Li, 2001)	$\begin{cases} \mu\alpha & \alpha \leq \mu \\ \frac{-\alpha^2+2\theta\mu\alpha-\mu^2}{2(\theta-1)} & \mu < \alpha \leq \theta\mu \\ \frac{\mu^2(1+\theta)}{2} & \alpha > \theta\mu \end{cases}$	$\begin{cases} \mu & \alpha \leq \mu \\ \frac{-\alpha+\theta\mu}{\theta-1} & \mu < \alpha \leq \theta\mu \\ 0 & \alpha > \theta\mu \end{cases}$	μ	$\frac{1}{\theta-1}$

the reuse of existing optimization algorithms originally designed for convex regularizers on these problems with nonconvex regularizers.

Similar results can be obtained for the g's in **C2** and **C3**.

Proposition 2.3. *For case* **C2***, g can be decomposed as $\bar{g}(x) + \breve{g}(x)$, where $\bar{g}(x) \equiv \sum_{i=1}^{d} \kappa(|x_i|) - \kappa_0\|x\|_1$ is concave and Lipschitz smooth, while $\breve{g}(x) \equiv \kappa_0\|x\|_1$ is convex and nonsmooth.*

Proposition 2.4. *For case* **C3***, g can be decomposed as $\bar{g}(X) + \breve{g}(X)$, where $\bar{g}(X) \equiv \sum_{i=1}^{m} \kappa(\sigma_i(X)) - \kappa_0\|X\|_*$ is concave and Lipschitz smooth, while $\breve{g}(X) \equiv \kappa_0\|X\|_*$ is convex and nonsmooth.*

The following shows that the critical points of (4) are also critical points of (1). This justifies learning via the reformulation in (4).

Proposition 2.5. *If x_* is a critical point of* (4)*, it is also a critical point of* (1)*.*

Recall that $\bar{g}$ is concave and $\breve{g}$ is convex. Hence, the nonconvex regularizer g is decomposed as a difference of convex functions (DC) (Hiriart-Urruty, 1985). Lu (2012) and Gong et al. (2013) also relied on DC decompositions of the regularizer. However, they do not utilize this in the computational procedures, and can only handle simple nonconvex regularizers. On the other hand, we use the DC decomposition to simplify the regularizers. As will be seen, while the DC decomposition of a nonconvex function is not unique, the particular one proposed here is crucial for efficient optimization. In the following, we provide concrete examples to show how the proposed convexification scheme can be used with various optimization algorithms originally designed for convex regularizers.

2.2. Usage with Proximal Algorithms

In this section, we provide example applications on using the proximal algorithm for structured sparse learning.

We will focus on several group lasso variants, though the proposed procedure can also be used in other applications involving proximal algorithms, such as fused lasso (Tibshirani et al., 2005) and graphical lasso (Jacob et al., 2009).

The proximal algorithm (Parikh & Boyd, 2013) has been commonly used for learning with convex regularizers. With a nonconvex regularizer, the underlying proximal step becomes much more challenging. It can still be solved with the concave-convex procedure or its variant sequential convex programming (SCP) (Lu, 2012). However, they are slow in general (Gong et al., 2013; Li & Lin, 2015), as will also be empirically demonstrated in Section 3.

2.2.1. NONCONVEX SPARSE GROUP LASSO

The feature vector x is divided into groups, and $\mathcal{G}_j$ contains dimensions in x that group j contains. Let $\left[x_{\mathcal{G}_j}\right]_i = x_i$ if $i \in \mathcal{G}_j$, and 0 otherwise. The (convex) sparse group lasso is formulated as (Jacob et al., 2009):

$$\min_x \sum_{i=1}^{N} \ell(y_i, a_i^\top x) + \lambda\|x\|_1 + \sum_{j=1}^{K} \mu_j\|x_{\mathcal{G}_j}\|_2,$$

where $\{(a_1, y_1), \ldots, (a_N, y_N)\}$ are the training samples, ℓ is a smooth loss, and K is the number of (non-overlapping) groups.

For the nonconvex extension, the regularizer becomes $g(x) = \lambda\sum_{i=1}^{d} \kappa(|x_i|) + \sum_{j=1}^{K} \mu_j\kappa\left(\|x_{\mathcal{G}_j}\|_2\right)$. Using Corollary 2.2 and Proposition 2.3, the convexified regularizer is then $\breve{g}(x) = \kappa_0(\lambda\|x\|_1 + \sum_{j=1}^{K} \mu_j\|x_{\mathcal{G}_j}\|_2)$. This can be easily handled by the proximal gradient algorithm in (Yuan et al., 2011). In particular, the proximal operator of $\breve{g}$ can be efficiently obtained by computing $\text{prox}_{\mu_j\|\cdot\|_2}\left(\text{prox}_{\lambda\|\cdot\|_1}\left(x_{\mathcal{G}_j}\right)\right)$ for each group separately.

As mentioned in Section 2.1, the DC decomposition is not unique. For example, we may decompose the nonconvex $g_i(x) = \kappa(\|x_{\mathcal{G}_i}\|_2)$ as $\breve{g}_i(x) + \bar{g}_i(x)$, where $\bar{g}_i(x) =$

$-\frac{\rho}{2}\|x_{\mathcal{G}_i}\|_2^2$ is concave and $\breve{g}_i(x) = \kappa\left(\|x_{\mathcal{G}_i}\|_2\right) + \frac{\rho}{2}\|x_{\mathcal{G}_i}\|_2^2$ is convex but nonsmooth. However, such a $\breve{g}_i(x)$ cannot be easily handled by existing proximal algorithms.

2.2.2. NONCONVEX TREE-STRUCTURED GROUP LASSO

In the (convex) tree-structured group lasso (Liu & Ye, 2010; Jenatton et al., 2011), dimensions in x are organized as nodes in a tree, and each group corresponds to a subtree.[2] The regularizer is of the form $\sum_{j=1}^K \mu_j \|x_{\mathcal{G}_j}\|_2$.

For the nonconvex extension, $g(x)$ becomes $\sum_{j=1}^K \mu_j \kappa\left(\|x_{\mathcal{G}_j}\|_2\right)$, and the convexified regularizer is $\breve{g}(x) \equiv \kappa_0 \sum_{j=1}^K \mu_j \|x_{\mathcal{G}_j}\|_2$. As shown in (Liu & Ye, 2010), its proximal step can be computed efficiently by processing all the groups once in some appropriate order.

2.3. Usage with the Frank-Wolfe Algorithm

The Frank-Wolfe (FW) algorithm (Jaggi, 2013), has recently been popularly used for many convex optimization problems in machine learning. In this section, we consider as an example the learning of low-rank matrices. Its optimization problem is of the form

$$\min_{X \in \mathbb{R}^{m \times n}} f(X) + \mu \sum_{i=1}^m \kappa(\sigma_i(X)), \qquad (5)$$

where f is the loss. For example, in matrix completion (Candès & Recht, 2009),

$$f(X) = \frac{1}{2}\|\mathcal{P}_\Omega(X - O)\|_F^2, \qquad (6)$$

where O is the observed incomplete matrix, $\Omega \in \{0,1\}^{m \times n}$ contains indices to the observed entries in O, and $[P_\Omega(A)]_{ij} = A_{ij}$ if $\Omega_{ij} = 1$; and 0 otherwise. When κ is the identity function, (5) reduces to the standard nuclear norm regularizer.

Using Proposition 2.4, it can be easily seen that after convexification, problem (5) can be rewritten as

$$\min_{X \in \mathbb{R}^{m \times n}} \bar{f}(X) + \mu\kappa_0\|X\|_*, \qquad (7)$$

where $\bar{f}(X) = f(X) + \bar{g}(X)$, and $\bar{g}(X) = \mu \sum_{i=1}^m \left(\kappa(\sigma_i(X)) - \kappa_0\sigma_i(X)\right)$. At the tth iteration of the FW algorithm, the key linear subproblem is $\min_{S:\|S\|_* \leq 1}\langle S, \nabla \bar{f}(X_t)\rangle$, where X_t is the current iterate. Its optimal solution can be easily obtained from the rank-one SVD of $\nabla \bar{f}(X_t)$ (Jaggi, 2013).

In contrast, the FW algorithm cannot be used directly on (5), as its linear subproblem then becomes

$\min_{S:\sum_{i=1}^m \kappa(\sigma_i(S))\leq 1}\langle S, \nabla f(X_t)\rangle$, which is difficult. Using other DC decompositions, such as $\bar{g}(X) = -\frac{\rho}{2}\|X\|_F^2$ and $\breve{g}(X) = \sum_{i=1}^m \kappa(\sigma_i(X)) + \frac{\rho}{2}\|X\|_F^2$, will not make the optimization easier. The linear subproblem in FW then becomes $\min_{S:\breve{g}(S)\leq 1}\langle S, \nabla \bar{f}(X_t)\rangle$, which is still difficult.

Though it is computationally feasible to use FW to solve (7), note that this transformed problem is nonconvex (because of $\bar{f}$), and convergence of the FW algorithm has only been shown for convex problems.

In the following, we propose a FW variant (Algorithm 1) for nonconvex problems of the form (5). A low-rank factorization $U_t B_t V_t^\top$ of X_t is maintained throughout the iterations. As in (Zhang et al., 2012), we adopt a local optimization scheme to speed up convergence. However, as $\bar{f}$ is nonconvex and the gradient of $\bar{g}$ depends on the singular values of X, the method in (Zhang et al., 2012) cannot be directly used. Instead, recall that the singular values are orthogonally invariant. Given U and V (the orthogonal left and right subspaces of X), we have $\bar{g}(X) = \bar{g}(UBV^\top) = \bar{g}(B)$ and $\|X\|_* = \|UBV^\top\|_* = \|B\|_*$. Thus, (7) can be rewritten as

$$\min_{U,B,V} \quad f(UBV^\top) + \bar{g}(B) + \mu\kappa_0\|B\|_* \quad (8)$$
$$\text{s.t.} \quad U^\top U = I, V^\top V = I,$$

where I is the identity matrix. This can be efficiently solved using the matrix optimization techniques on Grassmann manifold (Ngo & Saad, 2012).

Algorithm 1 Frank-Wolfe algorithm for solving (7). Here, QR denotes the QR factorization.

1: $U_0 = [], V_0 = []$;
2: **for** $t = 1 \ldots T$ **do**
3: $\quad [u_t, s_t, v_t] = \text{rank1SVD}(\nabla \bar{f}(X_t))$;
4: $\quad \bar{U}_t = \text{QR}([U_{t-1}, \sqrt{s_t}u_t])$;
5: $\quad \bar{V}_t = \text{QR}([V_{t-1}, \sqrt{s_t}v_t])$;
6: $\quad$ obtain $[U_t, B_t, V_t]$ from (8), using $\bar{U}_t, \bar{V}_t$ as warm-start; // ($X_t = U_t B_t V_t^\top$)
7: $\quad [\hat{U}, \Sigma_t, \hat{V}] = \text{SVD}(B_t)$;
8: $\quad U_t = U_t\hat{U}, V_t = V_t\hat{V}$; // ($X_t = U_t \Sigma_t V_t^\top$)
9: **end for**
output $X_t = U_t V_t^\top$.

Existing analysis for the FW algorithm are only for convex problems. The following theorem shows convergence of Algorithm 1 to a critical point of (5).

Theorem 2.6. *Assume that $X_t \neq 0$ for $t \geq 1$. The $\{X_t\}$ sequence generated by Algorithm 1 converges to a critical point of* (5).

Further speedup is possible in the special case of matrix completion problems (where f is given by (6)). Note from steps 7 and 8 of Algorithm 1 that X_t is implicitly stored as

a low-rank factorization $U_t \Sigma V_t^\top$. Using (6), we can obtain $\nabla \bar{f}(X_t)$ as (Watson, 1992)

$$\nabla \bar{f}(X_t) = \mathcal{P}_\Omega \left(U_t \Sigma_t V_t^\top - O \right) + U_t \hat{\Sigma} V_t^\top, \quad (9)$$

where $\hat{\Sigma} = [\hat{\Sigma}_{ii}]$ is a diagonal matrix with $\hat{\Sigma}_{ii} = \mu \left(\kappa'([\Sigma_t]_{ii}) - \kappa_0 \right)$. Note that Ω is sparse and the rank of X_t cannot be larger than t at the tth iteration. Thus, (9) admits a "sparse plus low-rank" structure, which can be used to significantly speed up the SVD computation (Mazumder et al., 2010).

2.4. Other Uses of the Proposed Scheme

The proposed scheme can also be used to simplify and speed up other nonconvex optimization problems in machine learning. Here, we consider as an example a recent nonconvex generalization of lasso (Gong & Ye, 2015), in which the standard ℓ_1 regularizer is extended to the nonconvex version $g(x) = \sum_{i=1}^d \kappa(|x_i|)$. Plugging this into (1), we arrive at the problem

$$\min_x f(x) + \mu \sum_{i=1}^d \kappa(|x_i|). \quad (10)$$

Gong & Ye (2015) proposed a sophisticated algorithm (HONOR) which involves a combination of quasi-Newton and gradient descent steps. Though the algorithm is similar to OWL-QN (Andrew & Gao, 2007) and its variant mOWL-QN (Gong, 2015), the convergence analysis in (Gong, 2015) cannot be directly applied as the regularizer is nonconvex. Instead, a non-trivial extension was developed in (Gong & Ye, 2015).

Here, by convexifying the nonconvex regularizer, (10) can be rewritten as

$$\min_x \bar{f}(x) + \mu \kappa_0 \|x\|_1, \quad (11)$$

where $\bar{f}(x) = f(x) + \bar{g}(x)$ and $\bar{g}(x) = \mu \sum_{i=1}^d (\kappa(|x_i|) - \kappa_0 |x_i|)$. It is easy to see that the convergence analysis for mOWL-QN (specifically, Propositions 4 and 5 in (Gong, 2015)) can be immediately applied, and guarantees convergence of mOWL-QN to a critical point of (11). By Proposition 2.5, this is also a critical point of (10). Moreover, as is demonstrated in previous sections, using other DC decompositions of g will not lead to the ℓ_1-regularizer in (11), and mOWL-QN can no longer be applied.

Problem (10) can be solved by either (i) directly using HONOR, or (ii) using mOWL-QN on the transformed problem (11). We believe that the latter approach is computationally more efficient. Note that both HONOR and mOWL-QN rely heavily on second-order information. In (11), the Hessian depends only on $\bar{f}$, as the Hessian

due to $\|x\|_1$ is zero (Andrew & Gao, 2007). However, in (10), the Hessian depends on both terms in the objective, as the second-order derivative of κ is not zero in general. HONOR constructs the approximate Hessian using only information from f, and thus ignores the curvature information due to $\sum_{i=1}^d \kappa(|x_i|)$. Hence, optimizing (11) with mOWL-QN is potentially faster, as all the second-order information is utilized. This will be verified empirically in Section 3.4.

3. Experiments

In this section, we perform experiments on using the proposed procedure with (i) proximal algorithms (Sections 3.1 and 3.2); (ii) Frank-Wolfe algorithm (Section 3.3); and (iii) comparision with HONOR (Section 3.4).

3.1. Nonconvex Sparse Group Lasso

In this section, we perform experiments on the nonconvex sparse group lasso model (Section 2.2.1)

$$\min_x \frac{1}{2} \|y - A^\top x\|_2^2 + \lambda \sum_{i=1}^d \kappa(|x_i|) + \mu \sum_{j=1}^K \kappa(\|x_{\mathcal{G}_j}\|_2), \quad (12)$$

where $\kappa(\cdot)$ is the LSP regularizer in Table 1 (with $\theta = 0.5$). The synthetic data set is generated as follows. The ground-truth parameter vector $\bar{x} \in \mathbb{R}^{10000}$ is divided into 100 non-overlapping groups: $\{1, \ldots, 100\}$, $\{101, \ldots, 200\}$, $\ldots$, $\{9901, \ldots, 10000\}$. We randomly set 75% of the groups to zero. In each nonzero group, we randomly set 25% of its features to zero, and generate the nonzero features from the standard normal distribution $\mathcal{N}(0, 1)$. Using 20000 samples, entries of the input matrix $A \in \mathbb{R}^{10000 \times 20000}$ are generated from $\mathcal{N}(0, 1)$. The ground-truth output is $\bar{y} = A^\top \bar{x}$, and training set output is $y = \bar{y} + \epsilon$, where ϵ is random noise following $\mathcal{N}(0, 0.05)$.

The proposed algorithm will be called N2C (Nonconvex-to-Convex). The proximal step of the convexified regularizer is obtained as in (Yuan et al., 2011), and the nonmonotonous accelerated proximal gradient algorithm (Li & Lin, 2015) is used for optimization. It will be compared with the following state-of-the-art algorithms:

1. SCP: Sequential convex programming (Lu, 2012), in which the LSP regularizer is decomposed as in Table 1 of (Gong et al., 2013).

2. GIST (Gong et al., 2013): Since the nonconvex regularizer is not separable, the associated proximal operator has no closed-form solution. Instead, we use SCP (with warm-start) to solve it numerically.

3. GD-PAN (Zhong & Kwok, 2014): It performs gradient descent with proximal average (Bauschke

Table 2. Results on nonconvex sparse group lasso. RMSE and MABS are scaled by 10^{-3}, and the CPU time is in seconds. The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

	non-accelerated			accelerated		convex
	SCP	GIST	GD-PAN	nmAPG	N2C	FISTA
RMSE	**50.6±2.0**	**50.6±2.0**	52.3±2.0	**50.6±2.0**	**50.6±2.0**	53.8±1.7
MABS	**5.7±0.2**	**5.7±0.2**	7.1±0.4	**5.7±0.2**	**5.7±0.2**	10.6±0.3
CPU time (sec)	0.84±0.14	0.92±0.12	0.94±0.22	0.65±0.06	**0.48±0.05**	0.79±0.14

Table 3. Results on tree-structured group lasso. The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

	SCP	GIST	GD-PAN	nmAPG	N2C	FISTA
testing accuracy (%)	**99.6±0.9**	**99.6±0.9**	**99.6±0.9**	**99.6±0.9**	**99.6±0.9**	97.2±1.8
sparsity (%)	5.5±0.4	5.7±0.4	6.9±0.4	5.4±0.3	**5.1±0.2**	9.2±0.2
CPU time (sec)	7.1±1.6	50.0±8.1	14.2±2.6	3.8±0.4	**1.9±0.3**	1.0±0.4

et al., 2008) of the nonconvex regularizers. Closed-form solutions for the proximal operator of each regularizer are obtained separately, and then averaged.

4. nmAPG (Li & Lin, 2015): This is the nonmontonous accelerated proximal algorithm. As for GIST, its proximal step does not have a closed-form solution and has to be solved numerically by SCP.

5. As a baseline, we also compare with the FISTA (Beck, 2009) algorithm, which solves the convex sparse group lasso model by removing κ from (12).

We do not compare with the concave-convex procedure (Yuille & Rangarajan, 2002), which has been shown to be slow (Gong et al., 2013; Zhong & Kwok, 2014). All algorithms are implemented in Matlab. The stopping criterion is reached when the relative change in objective is smaller than 10^{-8}. Experiments are performed on a PC with Intel i7 CPU and 32GB memory.

50% of the data are used for training, another 25% for validation and the rest for testing. We use a fixed stepsize of $\eta = \sigma_{\max}(A^\top A)$, while λ, μ in (12) are tuned by the validation set. For performance evaluation, we use the (i) testing root-mean-squared error (RMSE) on the predictions; (ii) mean absolute error of the obtained parameter $\hat{x}$ with ground-truth x, MABS $= \|\hat{x} - x\|_1/10000$; and (iii) CPU time. Each experiment is repeated 5 times, and the average performance reported.

Results are shown in Table 2, As can be seen, all the nonconvex models obtain better RMSE and MABS than FISTA, and N2C is the fastest. Note that GD-PAN is solving an approximate problem in each iteration, and its error is slightly worse than those of the other nonconvex algorithms on this data set. Figure 1 shows convergence of the objective with time for a typical run. Clearly, N2C is the fastest, as it is based on the accelerated proximal algorithm

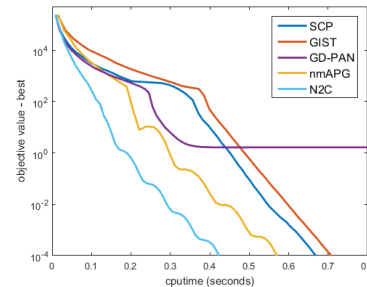with an inexpensive proximal step. GD-PAN, on the other hand, converges to an inferior solution.

Figure 1. Convergence of objective vs CPU time on nonconvex sparse group lasso. FISTA is not shown as its (convex) objective is different from the others.

3.2. Nonconvex Tree-Structured Group Lasso

In this section, we perform experiments on the nonconvex tree-structured group lasso model (Section 2.2.2). We use the face data set JAFFE[3], which contains 213 images with seven facial expressions: anger, disgust, fear, happy, neutral, sadness and surprise. Following (Liu & Ye, 2010), we resize each 256×256 image to 64×64. We also reuse their tree structure, which is based on pixel neighborhoods.

Since our goal is only to demonstrate usefulness of the proposed convexification scheme, we focus on the binary classification problem "anger vs not-anger". The logistic loss is used, which is more appropriate for classification. The optimization problem is

$$\min_x \sum_{i=1}^{N} w_i \log\left(1 + \exp\left(-y_i \cdot a_i^\top x\right)\right) + \sum_{i=1}^{K} \mu_i \kappa\left(\|x_{\mathcal{G}_i}\|_2\right),$$

where $\kappa(\cdot)$ is the LSP regularizer (with $\theta = 0.5$),

---
[3]http://www.kasrl.org/jaffe.html

Table 4. Results on the MovieLens data sets (CPU time is in seconds). The best RMSE's (according to the pairwise t-test with 95% confidence) are highlighted.

	MovieLens-100K			MovieLens-1M			MovieLens-10M		
	RMSE	rank	time	RMSE	rank	time	RMSE	rank	time
N2C-FW	**0.855±0.004**	2	< 1	**0.785±0.001**	5	**9.3±0.1**	**0.778±0.001**	9	**313.0±6.6**
FaNCL	0.857±0.003	2	< 1	0.786±0.001	5	16.6±0.6	0.779±0.001	9	615.7±13.2
LMaFit	0.867±0.004	2	< 1	0.812±0.002	5	14.7±0.7	0.797±0.001	9	491.9±36.3
active	0.875±0.002	52	1.8±0.1	0.811±0.001	106	46.3±1.1	0.808±0.001	137	1049.8±43.2

$\{(a_1, y_1), \ldots, (a_N, y_N)\}$ are the training samples, and w_i's are weights (set to be the reciprocal of the size of sample i's class) used to alleviate class imbalance. 60% of the data are used for training, 20% for validation and the rest for testing.

For the proposed N2C algorithm, the proximal step of the convexified regularizer is obtained as in (Liu & Ye, 2010). As in Section 3.1, it is compared with SCP, GIST, GD-PAN, nmAPG, and FISTA. The stepsize η is obtained by line search. For performance evaluation, we use (i) the testing accuracy; (ii) solution sparsity; and (iii) CPU time. Each experiment is repeated five times, and the performance averaged.

Results are shown in Table 3. As can be seen, all nonconvex models have similar testing accuracies, and they again outperform the convex model. Solutions from the nonconvex models are also sparser, with the GD-PAN solution being slightly denser due to its underlying approximation during optimization. The N2C solution is the sparsest. Moreover, N2C is the fastest, as can be seen from the convergence plot in Figure 2.
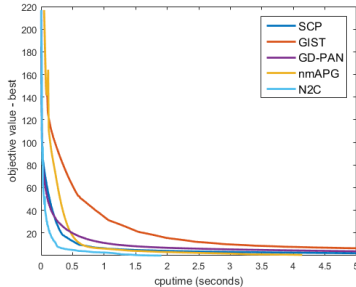

Figure 2. Convergence of objective vs CPU time on nonconvex tree-structured group lasso.

3.3. Nonconvex Low-Rank Matrix Completion

In this section, we perform experiments on matrix completion with the loss function in (6). The LSP regularizer is used, with $\theta = \sqrt{\mu}$ as in (Yao et al., 2015). We use the MovieLens data sets[4] (Table 5), which have

---
[4]http://grouplens.org/datasets/movielens/

been commonly used for evaluating matrix completion (Hsieh & Olsen, 2014; Yao et al., 2015). They contain ratings $\{1, 2, \ldots, 5\}$ assigned by various users on movies.

Table 5. MovieLens data sets used in the experiment.

	#users	#items	#ratings
100K	943	1,682	100,000
1M	6,040	3,449	999,714
10M	69,878	10,677	10,000,054

The proposed procedure (Algorithm 1), denoted N2C-FW, is compared with another nonconvex matrix regularization algorithm FaNCL recently proposed in (Yao et al., 2015). It is an efficient proximal algorithm using approximate SVD and automatic thresholding of singular values. The following two popular approaches are also used as baselines:

1. LMaFit (Wen et al., 2012): It factorizes X as a product of low-rank matrices $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$. The nonconvex objective $\frac{1}{2}\|P_\Omega(UV^\top - O)\|_F^2$ is then minimized by alternating minimization on U and V using gradient descent.

2. Active subspace selection (denoted "active") (Hsieh & Olsen, 2014): This solves the (convex) nuclear norm regularized problem (with κ being the identity function in (5)) by using the active row/column subspaces to reduce the optimization problem size.

Following (Yao et al., 2015), we use 50% of the ratings for training, 25% for validation and the rest for testing. The stopping criterion is reached when the relative change in objective is smaller than 10^{-4}. For performance evaluation, we use (i) the testing RMSE; and (ii) recovered rank.

Results are shown in Table 4. As can be seen, the convex model needs a much higher rank than the nonconvex ones, which agrees with the previous observations in (Mazumder et al., 2010; Yao et al., 2015). Figure 3 shows convergence of the objective with CPU time. As the recovered matrix rank is very low (2 to 9 in Table 4), N2C-FW is much faster than the others as it starts from a rank-one matrix and only
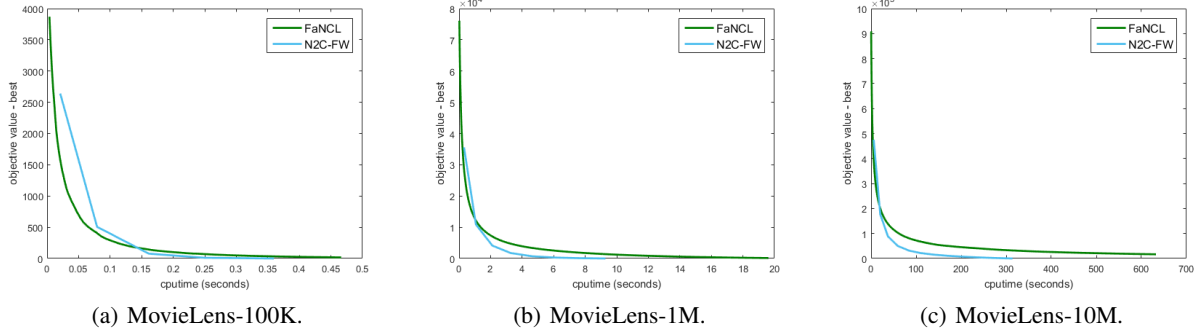
(a) MovieLens-100K.　　　　　　(b) MovieLens-1M.　　　　　　(c) MovieLens-10M.

Figure 3. Convergence of objective vs CPU time on nonconvex low-rank matrix completion. The other two baselines, LMaFit and active subspace selection, use different objectives and are thus not shown.
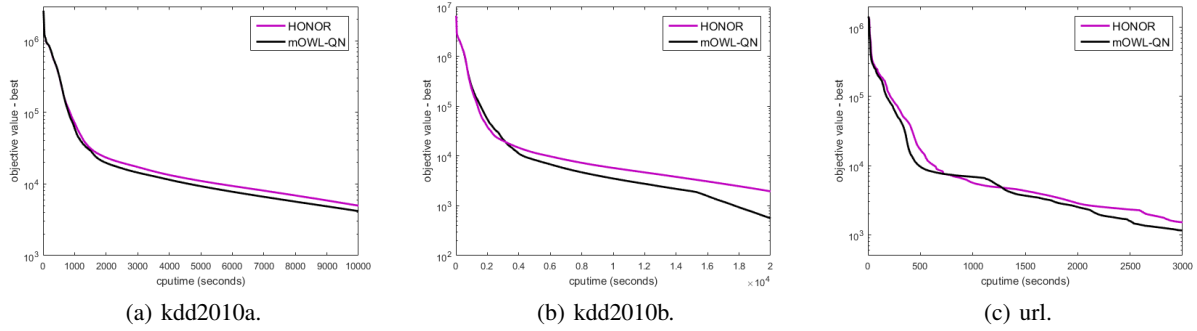

(a) kdd2010a.　　　　　　(b) kdd2010b.　　　　　　(c) url.

Figure 4. Convergence of objective vs CPU time for HONOR and mOWL-QN.

increases its rank by one in each iteration. Though FaNCL uses singular value thresholding to truncate the SVD, it does not control the rank as directly as N2C-FW and so is still slower.

3.4. Comparison with HONOR

In this section, we perform experiments on the model in (10), using the logistic loss and LSP regularizer. Following (Gong & Ye, 2015), we fix $\mu = 1$ in (10), and θ in the LSP regularizer to 0.01μ. Experiments are performed on three large data sets, "kdd2010a", "kdd2010b" and "url"[5] (Table 6). Both "kdd2010a" and "kdd2010b" are educational data sets, and the task is to predict students' successful attempts to answer concepts related to algebra. The "url" data set contains a collection of websites, and the task is to predict whether a particular website is malicious.

Table 6. Data sets used in the comparison with HONOR.

	kdd2010a	kdd2010b	url
#samples	510,302	748,401	2,396,130
#features	20,216,830	29,890,095	3,231,961

[5] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

We compare running HONOR (Gong & Ye, 2015) directly on (10) with running mOWL-QN (Gong, 2015)) on the transformed problem (11). In HONOR, the threshold of the hybrid step is set to 10^{-10}, which yields the best empirical performance in (Gong & Ye, 2015).

Figure 4 shows convergence of the objective with CPU time (note that the objectives of (10) and (11) are equivalent). As can be seen, mOWL-QN converges faster than HONOR, which validates our claim that the curvature information of the nonconvex regularizer helps.

4. Conclusion

In this paper, we proposed a novel approach to learning with nonconvex regularizers. By moving the nonconvexity associated with the nonconvex regularizer to the loss, the nonconvex regularizer is convexified to become a familiar convex regularizer while the augmented loss is still Lipschitz smooth. This allows one to reuse efficient algorithms originally designed for convex regularizers on the transformed problem. In particular, we illustrate how this can be used with the proximal algorithm and Frank-Wolfe algorithm. Experiments on a number of nonconvex regularization problems show that the proposed procedure is much faster than the state-of-the-art.

Acknowledgments

References

Andrew, G. and Gao, J. Scalable training of ℓ_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine learning*, pp. 33–40, 2007.

Bauschke, Heinz H, Goebel, Rafal, Lucet, Yves, and Wang, Xianfu. The proximal average: basic theory. *SIAM Journal on Optimization*, 19(2):766–785, 2008.

Beck, A.and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Candès, E.J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Candès, E.J., Wakin, M.B., and Boyd, S.P. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Geman, D. and Yang, C. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995.

Gong, P.and Ye, J. A modified orthant-wise limited memory quasi-Newton method with convergence analysis. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 276–284, 2015.

Gong, P. and Ye, J. HONOR: Hybrid Optimization for NOn-convex Regularized problems. In *Advances in Neural Information Processing Systems*, pp. 415–423, 2015.

Gong, P., Zhang, C., Lu, Z., Huang, J., and Ye, J. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proceedings of the International Conference on Machine Learning*, pp. 37–45, 2013.

Hiriart-Urruty, J.B. Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. 1985.

Hsieh, C.-J. and Olsen, P. Nuclear norm minimization via active subspace selection. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 575–583, 2014.

Jacob, L., Obozinski, G., and Vert, J.-P. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 433–440, 2009.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 427–435, 2013.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

Li, H. and Lin, Z. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems*, pp. 379–387, 2015.

Liu, J. and Ye, J. Moreau-Yosida regularization for grouped tree structure learning. In *Advances in Neural Information Processing Systems*, pp. 1459–1467, 2010.

Liu, J., Musialski, P., Wonka, P., and Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

Loh, P.-L. and Wainwright, M.J. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pp. 476–484, 2013.

Lu, Z. Sequential convex programming methods for a class of structured nonlinear programming. Preprint arXiv:1210.3039, 2012.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

Ngo, T. and Saad, Y. Scaled gradients on Grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, pp. 1412–1420, 2012.

Parikh, N. and Boyd, S.P. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1): 91–108, 2005.

Trzasko, J. and Manduca, A. Highly undersampled magnetic resonance image reconstruction via homotopic-minimization. *IEEE Transactions on Medical Imaging*, 28(1):106–121, 2009.

Watson, G.A. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Wen, Z., Yin, W., and Zhang, Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

Yao, Q., Kwok, J.T., and Zhong, W. Fast low-rank matrix learning with nonconvex regularization. In *Proceedings of IEEE International Conference on Data Mining*, pp. 539–548, 2015.

Yuan, L., Liu, J., and Ye, J. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems*, pp. 352–360, 2011.

Yuille, A.L. and Rangarajan, A. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, pp. 1033–1040, 2002.

Zhang, C.H. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2): 894–942, 2010a.

Zhang, T. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

Zhang, X., Schuurmans, D., and Yu, Y.-L. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems*, pp. 2906–2914, 2012.

Zhong, W. and Kwok, J.T. Gradient descent with proximal average for nonconvex and composite regularization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 2206–2212, 2014.