Efficient Learning with a Family of Nonconvex Regularizers by Redistributing Nonconvexity

Quanming Yao  James T. Kwok  {qyaoaa, jamesk}@cse.ust.hk
Department of Computer Science and Engineering

**Background**

Machine learning objective
\[
\min F(x) \equiv f(x) + g(x)
\]
- \(f\): loss function; \(g\): regularizer
  - e.g., \(\ell_1\) norm: image denoising
  - e.g., \(\|x\|_1\): nuclear norm: collaborative filtering

Popular nonconvex regularizers
- GP: Geman penalty
- Laplace penalty
- LSP: log-sum penalty
- MCP: minmax concave penalty
- SCAD: smoothly clipped absolute deviation penalty

<table>
<thead>
<tr>
<th>optimization</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>performance</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
</tbody>
</table>

Can we have the best of both worlds?

**Idea**

- a novel scheme to decompose nonconvex regularizers
  - \(f\): smooth  \(g\): nonconvex

\[\hat{f}: \text{smooth} \quad \hat{g}: \text{convex}\]

Transform \(f + g \rightarrow \hat{f} + \hat{g}\)

Fast optimization & Good performance

Code is available at
- https://github.com/quanmingyao/N2C

**Assumptions**

Assumptions on (1)
- \(F\): bounded from below
- \(f\) (loss): Lipschitz-smooth
- \(g\) (regularizer): nonsmooth, nonconvex

Three types of nonconvex \(g\) will be considered

1. \(g(x) = \mu \sum_{i=1}^{d} \kappa(|x_i|)\)
   - \(\kappa()\) is concave, non-decreasing, \(\rho\)-Lipschitz smooth with \(\kappa(0) = 0\)
   - e.g., GP, Laplace, LSP, MCP, SCAD
   - \(\kappa(\cdot) = \alpha \rightarrow \text{lasso}\)

2. \(g(x) = \sum_{i=1}^{d} \mu_i \phi_i(x), \quad \phi_i(x) = \kappa(|A_i x|_p)\)
   - \(\mu_i \geq 0, A_i\) is a matrix, \(p \in (1, +\infty)\)
   - different \(A_i\)’s \(\rightarrow \text{group lasso, fused lasso, graphical lasso, etc.}\)

3. \(g(x) = \mu \sum_{i=1}^{d} \kappa_i \sigma_i(X)\)
   - \(X: \text{matrix}; \sigma_i(X): \text{singular values}\)
   - \(\kappa(\cdot) = \alpha \rightarrow \text{nuclear norm regularizer}\)

**Redistributing Nonconvexity**

Consider nonconvex lasso (Case C1). Rewrite \(g\) as
\[
g(x) = \mu \left( \sum_{i=1}^{d} \kappa(|x_i|) - \kappa_0 \|x\|_1 \right) + \mu_0 \|x\|_1, \quad \text{where} \quad \kappa_0 = \kappa(0)
\]

**Key observation** \(\kappa(|\cdot|) - \kappa_0 |\cdot|\) is smooth and concave

**Proposition (Case C1)** \(g(x)\) can be decomposed as \(\tilde{g}(x) + \hat{g}(x)\), where
- \(\tilde{g}(x) \equiv \mu \sum_{i=1}^{d} \kappa(|x_i|) - \kappa_0 |x|_1\) is concave and smooth
- \(\hat{g}(x) \equiv \mu_0 |x|_1\) is convex and nonsmooth

**Corollary (Case C2)** \(g(x)\) can be decomposed as \(\tilde{g}(x) + \hat{g}(x)\), where
- \(\tilde{g}(x) \equiv \sum_{i=1}^{K} \mu_i \phi_i(x)\) and \(\hat{g}(x) = \mu \kappa(|A_i x|_p) - \kappa_0 \mu_i |A_i x|_p\) is concave and Lipschitz smooth;
- \(\tilde{g}(x) \equiv \kappa_0 \sum_{i=1}^{K} \kappa_i \|A_i x\|_p\) is convex and nonsmooth

**Proposition (Case C3)** \(g(X)\) can be decomposed as \(\tilde{g}(X) + \hat{g}(X)\), where \(\tilde{g}(X) \equiv \sum_{i=1}^{d} \kappa_i \sigma_i(X)\), is concave and Lipschitz smooth; \(\hat{g}(X) \equiv \kappa_0 |X|_1\), is convex and nonsmooth

Nonconvex regularizer \(g\) is decomposed as a difference of convex functions. While the DC decomposition of a nonconvex function is not unique, the particular one proposed here is crucial for efficient optimization

- \(\hat{f}\) (augmented loss): smooth, nonconvex
- as both \(f\) and \(\hat{g}\) are smooth
- \(g\) (convex regularizer): convex, nonsmooth

Nonconvexity is shifted from regularizer to loss, while still ensuring that the augmented loss is smooth

Allows reuse of solvers for convex regularizers on nonconvex regularizers
- proximal algorithms - cheap proximal step
- Frank-Wolfe algorithm - cheap subproblem

Fast optimization & Good performance

**Example Use Cases**

With proximal algorithms: Sparse group lasso (with groups \(\mathcal{G}\))

\[
\begin{align*}
\text{original problem} & \quad \min f(x) + \lambda \sum_{i=1}^{K} \kappa_i (|x_i|) + \\
\text{transformed problem} & \quad \min_{\hat{f}} f(x) + \lambda \mu_0 \|x\|_1 + \\
\text{proximal step} & \quad \min_{\hat{g}} \frac{1}{2} \|x - \hat{g}\|_2^2 + \lambda \sum_{i=1}^{K} \kappa_i (|x_i|) + \\
& \quad \min_{\hat{g}} \frac{1}{2} \|x - \hat{g}\|_2^2 + \lambda \sum_{i=1}^{K} \kappa_i (|x_i|) + \\
& \quad \text{difficult to solve} \quad \text{cheap closed-form solution} \quad (Yuan et al., 2011)
\end{align*}
\]

With Frank-Wolfe algorithm: Low-rank matrix learning

\[
\begin{align*}
\text{original problem} & \quad \min_{X} f(X) + \mu \sum_{i=1}^{K} \kappa_i (\sigma_i(X)) + \\
\text{transformed problem} & \quad \min_{\hat{f}} f(X) + \mu_0 \|X\|_* + \\
\text{subproblem} & \quad \min_{\hat{g}} \langle S, \nabla f(X) + \sum_{i=1}^{K} \kappa_i (\sigma_i(X)) \rangle \leq 1 + \\
& \quad \text{difficult to solve} \quad \text{cheap rank-1 SVD on} \quad \hat{g}(X) \quad (Zhang et al., 2012)
\end{align*}
\]

As \(\hat{f}\) is not convex, we also proposed a FW variant with convergence guarantees

**Experiments**

Sparse group lasso, we compare
- SCP, GIST, GD-PAN, nmAPG: algorithms on original problem; proximal steps in GIST & nmAPG are solved iteratively
- N2C: nmAPG on transformed problem (cheap closed-form solution for proximal step)
- FISTA on convex sparse problem

Experiments on synthetic data
- RMSE: 0.0506 (N2C) vs 0.0538 (FISTA)
- N2C is fast and accurate

Low-rank matrix completion:
\[
\min_{X} \frac{1}{2} \sum_{i,j \in \Omega}(X_{ij} - O_{ij})^2 + \mu \sum_{i=1}^{m} \kappa_i (\sigma_i(X))
\]
where \(O_{ij}\): observed ratings; \(\Omega\): set of observed entries. We compare
- active set method for nuclear norm regularized convex problem
- LMaFit: factorization approach
- FaNCL: inexact proximal algorithm on original problem
- N2C-FW: proposed FW algorithm on transformed problem

Experiments are performed on MovieLens-10M data
- RMSE: 0.778 (N2C) vs 0.797 (LMaFit) vs 0.808 (Active)
- N2C-FW is much faster than FaNCL