Active set strategy for high-dimensional non-convex sparse optimization problems

A. Boisbunon, R. Flamary, A. Rakotomamonjy

CNES/INRIA – AYIN Team
Université de Nice Sophia-Antipolis – Lab. Lagrange – OCA – CNRS
Université de Rouen – Lab. LITIS

May 7 2014
Motivation: high dimensional linear estimation

\[
\min_{x \in \mathbb{R}^p} \{ \ f(x) = l(x) + r(x) \ \} 
\]

\[
\begin{array}{c|c}
\text{Objective} & \text{Reg. term (nonconvex, nondifferentiable)} \\
\text{Data term} & \text{(differentiable)} \\
\end{array}
\]

**Objective**

- Estimate a high dimensional sparse model \( x \in \mathbb{R}^p \).
- Go beyond the Lasso (biased, not always consistent [8, 2]).
- Regularization term DC function:
  \[
  r(x) = \sum_{i=1}^{p} h(|x_i|)
  \]
- Use sparsity for efficient optimization.
- Build on top of existing efficient algorithms [6, 4].
Nonconvex sparse optimization in the literature

**Difference of Convex Algorithm (DCA) [1, 2, 3]**
- Solves iteratively weighted $\ell_1$-penalty.
- Slow but converges in few re-weighting operations.

**Sequential Convex Programming (SCP) [6]**
- Uses a majorization of the nonconvex penalty.
- Also handles constrained optimization.

**General Iterative Shrinkage and Threshold (GIST) [4]**
- Extension of proximal methods to nonconvex regularization.
- Estimation of descent step via BB-rule (Barzilai & Borwein).

**Limits of those approaches**
- Solve the full optimization problem.
- Full gradient computation is expensive.
  - Use an active set to focus on a small number of variables.
Active set strategy

Principle

▶ Work on a subset of variables $\varphi$ and solve the problem on this subset.
▶ Optimality conditions used to update the active set.
▶ Widely used in convex optimization.
▶ Sparse optimization: initialization $\varphi = \emptyset$.

Nonconvex optimality conditions

▶ The regularization term is expressed as a DC function:
  \[ r(x) = r_1(x) - r_2(x) \]
  with $r_1$ and $r_2$ two convex functions of the form
  \[ r_1(x) = \sum_i g_1(|x_i|), \quad r_2(x) = \sum_i g_2(|x_i|) \]  (2)

▶ If $x^*$ is a stationary point of the optimization problem then
  \[ \partial r_2(x^*) \subset \nabla l(x^*) + \partial r_1(x^*) \]  (3)
Optimality conditions in practice

Optimality conditions

- \( r(x) = \sum_i^p h(|x_i|) = \sum_i^p \{g_1(|x_i|) - g_2(|x_i|)\} \)
- Component-wise optimality condition.
- When \( g_2'(0) = 0 \) the optimality condition becomes
  \[ |\nabla l(x)_i| \leq g_1'(0) \quad \text{if} \quad x_i = 0. \]
- When \( g_2 = g_1 - h \) the optimality condition becomes
  \[ |\nabla l(x)_i| \leq h'(0) \quad \text{if} \quad x_i = 0. \]

Examples:

- \( \ell_1 : \quad h(u) = \lambda u \quad \Rightarrow \quad |\nabla l(x)_i| \leq \lambda \quad \text{if} \quad x_i = 0 \)
- Capped-\( \ell_1 : \quad h(u) = \lambda \min(u, \theta) \quad \Rightarrow \quad |\nabla l(x)_i| \leq \lambda \quad \text{if} \quad x_i = 0 \)
- Log sum : \( h(u) = \lambda \log(1 + u/\theta) \quad \Rightarrow \quad |\nabla l(x)_i| \leq \lambda/\theta \quad \text{if} \quad x_i = 0 \)
Active set algorithm

Algorithm for Log sum regularization

Inputs
- Initial active set $\varphi = \emptyset$

1: repeat
2: $x \leftarrow$ Solve Problem (1) with current active set $\varphi$ (using GIST)
3: Compute $r \leftarrow |\nabla l(x)|$
4: for $k = 1, \ldots, k_s$ do
5: $j \leftarrow \arg \max_{i \in \varphi} r_i$
6: If $r_j > h'(0) + \epsilon$ then $\varphi \leftarrow j \cup \varphi$
7: end for
8: until stopping criterion is met

Discussion

- Only small problems are solved (dimension $|\varphi|$).
- Use warm-starting trick.
- At each iteration, $k_s$ variables are added to the active set.
- Step 3 can be computed in parallel.
- $\epsilon > 0$ typically small, acts as a threshold similar to OMP.
Numerical experiments

Datasets

- Simulated Dataset: \( p = [10^2, 10^7] \), SNR=30dB, \( n = 100 \), \( t = 10 \).
- Dorothea Dataset: \( p = 10^5 \), \( n = 1150 \).
- URL Reputation Dataset: \( p = 3.2 \times 10^6 \), \( n = 20 000 \), sparse.

Compared Methods

- DC Algorithm, reweighted-\( \ell_1 \) (DC-Lasso) [2, 3].
- General Iterative Shrinkage and Threshold (GIST) [4].
- Proposed Active Set approach with GIST (AS-GIST).

Performance measures

- CPU time used in the algorithm.
- Final objective value.
  Both measures averaged over 10 splits/generations of the data.

Parameters

- Regularized least-squares.
- Log sum with \( \theta = 0.1 \).
- \( k_s = 10 \) and \( \epsilon = 0.1 \).
- Computed on Octave.
Simulated dataset

Results

- Standard deviation in dashed lines.
- DC-Lasso outperformed by GIST and AS-GIST.
- GIST and AS-GIST statistically equivalent and $\geq$ DC-Lasso.
- AS-GIST up to $20 \times$ faster than GIST and $\geq 100 \times$ faster than DC-Lasso.
Dorothea dataset

Results

- Performance measures along the regularization path.
- DC-Lasso not computed due to computational time.
- AS-GIST more efficient on sparse solutions (large $\lambda$).
- Better objective value of AS-GIST for small $\lambda$. 
URL Reputation dataset

Results

- Very high dimension $p = 3.2 \times 10^6$
- Important computational gain with AS-GIST.
- Important gain in objective value for small $\lambda$ ($\epsilon$ parameter).
**Conclusion**

**Active set strategy**
- When solution is sparse: use active set even for nonconvex problems.
- Spends more time optimizing values that count.
- Applicable to a wide class of regularization term.
- Any convex differentiable loss (least-squares, logistic regression).
- Simple algorithm, code will be available.

**Working on**
- More general optimality condition (Clarke differential).
- Convergence proof to stationary point.
- Study the regularization effect of initializing by 0 and choice of $\epsilon$.
- Applications in large scale datasets/problems.
The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems.  

Enhancing sparsity by reweighted $\ell_1$ minimization.  

Recovering sparse signals with a certain family of nonconvex penalties and DC programming.  

A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems.  
Result analysis of the NIPS 2003 feature selection challenge.

Sequential convex programming methods for a class of structured nonlinear programming.

Identifying suspicious URLs: an application of large-scale online learning.

The adaptive lasso and its oracle properties.
Examples of optimization problems

$$\min_{x \in \mathbb{R}^p} \{ f(x) = l(x) + r(x) \}$$

Data-fitting term

- Least-squares: $$l(x) = \frac{1}{2} \sum_k (y_i - a_k^\top x)^2 = \frac{1}{2} \| y - Ax \|^2$$
- Logistic regression: $$l(x) = \sum_k \log(1 + \exp(-y_k a_k^\top x))$$
- SVM Rank: $$l(x) = \sum_k \max(0, 1 - a_k^\top x)^2$$

Gradient of the form $$\nabla l(x) = A^\top e(x)$$

Regularization term

- Lasso ($\ell_1$): $$r(x) = \lambda \| x \|_1$$
- Capped-$\ell_1$: $$r(x) = \lambda \sum_i \min(|x_i|, \theta)$$
- Log sum: $$r(x) = \lambda \sum_i \log(1 + |x_i|/\theta)$$
- $\ell_p$-pseudonorm: $$r(x) = \lambda \sum_i |x_i|^p$$

Regularizer of the form $$r(x) = \sum_i^p h(|x_i|)$$