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

$$\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) = l(\mathbf{x}) + r(\mathbf{x}) \right\} \tag{1}$$

Objective

Data term (differentiable)

Reg. term (nonconvex, nondifferentiable)

Objective

- Estimate a high dimensional sparse model $\mathbf{x} \in \mathbb{R}^p$.
- Go beyond the Lasso (biased, not always consistent [8, 2]).
- Regularization term DC function:

$$r(\mathbf{x}) = \sum_i^p h(|x_i|)$$

- 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 ℓ_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.

$\rightarrow$ Use an active set to focus on a small number of variables.

Active set strategy

Principle

- Work on a subset of variables φ 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:
  $r(\mathbf{x}) = r_1(\mathbf{x}) - r_2(\mathbf{x})$ with r_1 and r_2 two convex functions of the form

$$r_1(\mathbf{x}) = \sum_i g_1(|x_i|), \quad r_2(\mathbf{x}) = \sum_i g_2(|x_i|) \tag{2}$$

- If $\mathbf{x}^*$ is a stationary point of the optimization problem then

$$\partial r_2(\mathbf{x}^*) \subset \nabla l(\mathbf{x}^*) + \partial r_1(\mathbf{x}^*) \tag{3}$$

Optimality conditions in practice

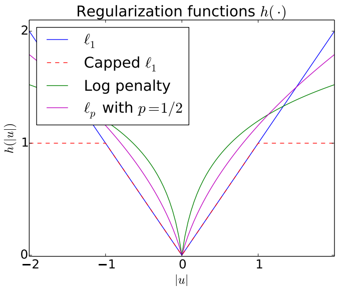
Optimality conditions

- $r(\mathbf{x}) = \sum_i^p h(|x_i|) = \sum_i^p \{g_1(|x_i|) - g_2(|x_i|)\}$
- Component-wise optimality condition.
- When $g_2'(0) = 0$ the optimality condition becomes

$$|\nabla l(\mathbf{x})_i| \leq g_1'(0) \quad \text{if} \quad x_i = 0.$$

- When $g_2 = g_1 - h$ the optimality condition becomes

$$|\nabla l(\mathbf{x})_i| \leq h'(0) \quad \text{if} \quad x_i = 0.$$



Examples:

$$\begin{aligned}
\ell_1: &\quad h(u) = \lambda u &\Rightarrow&\quad |\nabla l(\mathbf{x})_i| \leq \lambda \quad \text{if} \quad x_i = 0\\
\text{Capped-}\ell_1: &\quad h(u) = \lambda \min(u,\theta) &\Rightarrow&\quad |\nabla l(\mathbf{x})_i| \leq \lambda \quad \text{if} \quad x_i = 0\\
\text{Log sum}: &\quad h(u) = \lambda \log(1 + u/\theta) &\Rightarrow&\quad |\nabla l(\mathbf{x})_i| \leq \lambda/\theta \quad \text{if} \quad x_i = 0
\end{aligned}$$

Active set algorithm

Algorithm for Log sum regularization

Inputs

- Initial active set $\varphi = \emptyset$

1: **repeat**
2: $\quad$ $\mathbf{x} \leftarrow$ Solve Problem (1) with current active set φ (using GIST)
3: $\quad$ Compute $\mathbf{r} \leftarrow |\nabla l(\mathbf{x})|$
4: $\quad$ **for** $k = 1, \ldots, k_s$ **do**
5: $\quad\quad$ $j \leftarrow \arg\max_{i \in \bar{\varphi}} r_i$
6: $\quad\quad$ If $r_j > h'(0) + \varepsilon$ then $\varphi \leftarrow j \cup \varphi$
7: $\quad$ **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-ℓ_1 (DC-Lasso) [2, 3].
- General Iterative Shrinkage and Threshold (GIST) [4].
- Proposed Active Set approach with GIST (AS-GIST).

Performance measures

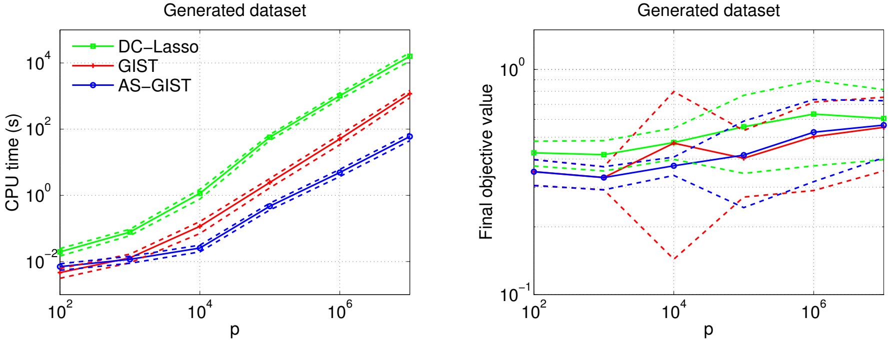
- CPU time used in the algorithm.
- Final objective value.

  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.

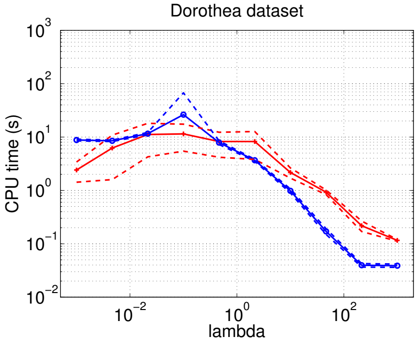
Simulated dataset

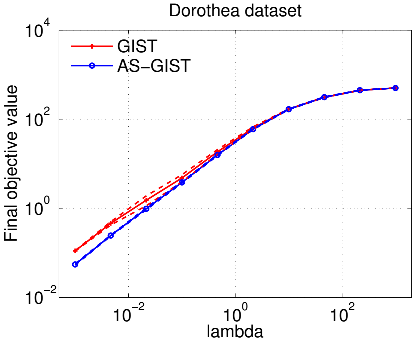


Results

- Standard deviation in dashed lines.
- DC-Lasso outperformed by GIST and AS-GIST.
- GIST and AS-GIST statistically equivalent and > DC-Lasso.
- AS-GIST up to 20× faster than GIST and > 100× faster than DC-Lasso.

Dorothea dataset

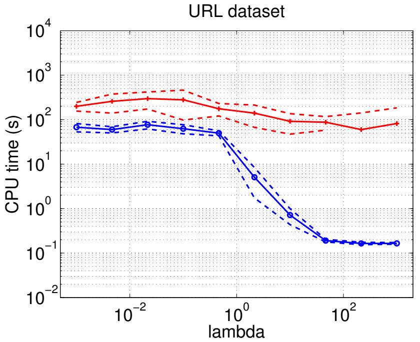


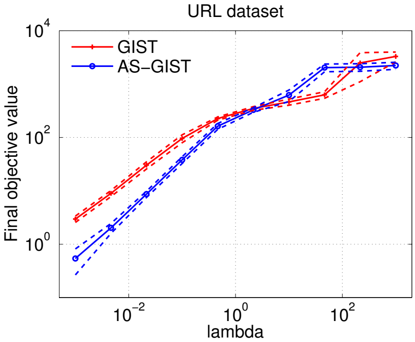


Results

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

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 λ (ϵ 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 $\mathbf{0}$ and choice of ϵ.
- Applications in large scale datasets/problems.

References I

[1] L.T.H. An and P.D. Tao.
The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems.
Annals of Operations Research, 133(1):23–46, 2005.

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

[3] G. Gasso, A. Rakotomamonjy, and S. Canu.
Recovering sparse signals with a certain family of nonconvex penalties and DC programming.
IEEE Transactions on Signal Processing, 57(12):4686–4698, 2009.

[4] P. Gong, C. Zhang, Z. Lu, and J. Huang, J. Ye.
A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems.
In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28, pages 37–45, 2013.

References II

[5] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror.
Result analysis of the NIPS 2003 feature selection challenge.
In *Advances in Neural Information Processing Systems*, pages 545–552, 2004.

[6] Z. Lu.
Sequential convex programming methods for a class of structured nonlinear programming.
arXiv preprint arXiv:1210.3039, 2012.

[7] J. Ma, L.K. Saul, S. Savage, and G.M. Voelker.
Identifying suspicious URLs: an application of large-scale online learning.
In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 681–688, 2009.

[8] H. Zou.
The adaptive lasso and its oracle properties.
Journal of the American Statistical Association, 101(476):1418–1429, 2006.

Examples of optimization problems

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

Data-fitting term

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

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

Regularization term

- Lasso (ℓ_1) : $r(\mathbf{x}) = \lambda \|\mathbf{x}\|_1$
- Capped-ℓ_1 : $r(\mathbf{x}) = \lambda \sum_i \min(|x_i|, \theta)$
- Log sum : $r(\mathbf{x}) = \lambda \sum_i \log(1 + |x_i|/\theta)$
- ℓ_p-pseudonorm : $r(\mathbf{x}) = \lambda \sum_i |x_i|^p$

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