Fast Active-set-type Algorithms for $L_1$-regularized Linear Regression

Jingu Kim and Haesun Park  
{jingu,hpark}@cc.gatech.edu  
School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology

Abstract

We present an efficient active-set-type algorithm for $L_1$-regularized linear regression, also known as the Lasso. Our new method, called block principal pivoting, accelerates computation by allowing exchanges of several variables among working sets. This method is significantly faster than existing active set methods and competitive against recently developed iterative methods.

Formulation and Dual

Training data: $(x_i, y_i)_{i=1}^n$, $y_i \in \mathbb{R}$ is the response of $x_i \in \mathbb{R}^p$. $y_i$'s are centered: $\bar{y}_i = 0$.

Primal:

$$\min_{\beta, \lambda} \mathcal{L}(\beta, \lambda) = \frac{1}{2} \|y - X\beta\|^2 + \lambda \|\beta\|_1,$$

(1)

Dual: for $r = y - X\beta$,

$$\min_{r} \mathcal{L}(r, \lambda) = \frac{1}{2} \|r\|^2 + \|y - X\beta\|_r \text{ s.t. } |X^Tr|_{\infty} \leq \lambda, \quad (2)$$

Optimality Conditions

Let $r^*$ is the solution of Eq. (2). The Karush-Kuhn-Tucker (KKT) conditions are

$$d^* = X^T r^* = X^T y - X^T X \beta^*,$$

(3a)

$$-\lambda \leq d^* \leq \lambda,$$

(3b)

$$-\lambda \leq d^*_i < \lambda \implies \beta^*_i = 0,$$

(3c)

$$d^*_i = \lambda \implies \beta^*_i \geq 0,$$

(3d)

$$d^*_i = -\lambda \implies \beta^*_i \leq 0,$$

(3e)

Active Set Methods

- Indices for active constraints: $\mathcal{E}_+ = \{ i | X^T r^*_i = \lambda \}$, $\mathcal{E}_- = \{ i | X^T r^*_i = -\lambda \}$.
- If $\mathcal{E}_+$ and $\mathcal{E}_-$ are known in advance, the solution $\beta^*$ can be easily computed by using Eq. (3c) and solving a normal equation Eq. (3a) for $\beta$.
- Maintain working sets ($\mathcal{E}_+, \mathcal{E}_-$) as candidates for $\mathcal{E}_+$ and iteratively update ($\mathcal{E}_+$, $\mathcal{E}_-$).
- Feature-sign search (Lee et al., 2007): Standard active set method derived for dual.
- LARS (Efron et al., 2004): Modified for full regularization path.
- Limitation: Only one variable is exchanged at each step, making many iterations for large problems.

Main Question

How can we efficiently identify active (and passive) constraints, $\mathcal{E}_+, \mathcal{E}_-$?

Block Principal Pivoting

Eqs. (3) are in a form of the linear complementarity problem with bounds (BLCP). An efficient way of finding $\mathcal{E}_+, \mathcal{E}_-$ for BLCP (Judice and Pereira, 1994):

1. Partition $\{1, \ldots, p\}$ into three disjoint subsets ($H, F, J$).
2. Assume: $\beta_H = 0$, $d_Y = \lambda$, $d_F = -\lambda$. (4)
3. Solve for $d_Y$ and compute $d_Y$ using
   $$d^*_Y = \begin{cases} X^T y & \text{if } d_F = \lambda, \\ X^T y - X^T \beta_F & \text{if } d_F < \lambda, \\ X^T y - X^T \beta_F & \text{if } d_F > \lambda, \end{cases}$$
   (5a)
   $$d^*_Y = \begin{cases} X^T y & \text{if } d_F = -\lambda, \\ X^T y - X^T \beta_F & \text{if } d_F < \lambda, \\ X^T y - X^T \beta_F & \text{if } d_F > \lambda, \end{cases}$$
   (5b)
4. If $-\lambda \leq d_Y < \lambda$, $\beta_F \geq 0$, $\beta_F \leq 0$, solution is found. If not, update $(H, F, J)$ and try again.
5. Infeasible variables: $\mathcal{G} = \bigcup_{J \neq J} J_i$ where $J_i = \{i \in H : d_i > \lambda\}$, $J_i = \{i \in H : d_i < -\lambda\}$, $J_i = \{i \in F : \beta_i < 0\}$, $J_i = \{i \in F : \beta_i > 0\}$.

- Take a subset $\mathcal{G} \subseteq \mathcal{G}$, Let $\hat{J} \subset \hat{J} \cap \mathcal{G}$.
- Update: $H \leftarrow (H - \{J_i \cup J\} \cup J \cup \{J\} \cup J\}$, (7a)
  $F \leftarrow (F - \{J_i \cup J\} \cup J \cup \{J\} \cup J\}$, (7b)
  $F \leftarrow (F - \{J_i \cup J\} \cup J \cup \{J\} \cup J\}$, (7c)
- $|\mathcal{G}| > 1$: block principal pivoting
- $|\mathcal{G}| = 1$: single principal pivoting (ex. active set)

Full exchange rule: $\mathcal{G} = G$

Finite termination: Use a backup rule when the full exchange rule does not work well:

$$\hat{G} = \{i : t = \max \{j : j \in \mathcal{G}\} \}.$$  

Algorithm

Input: $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$

Output: $\beta$
1. Initialize $H = \{1, \ldots, p\}$, $F = \emptyset$, $\beta = 0$.
2. $d = -X^T y$, $k = K_{max}$ and $\mathcal{G}$.
3. While $(\beta, d)$ is infeasible do
   4. Find $\mathcal{G}$ by Eqs. (6).
   5. If $|\mathcal{G}| < t$, set $\mathcal{G} = \mathcal{G}$, $k = K_{max}$.
   6. If $|\mathcal{G}| > t$ and $k > 1$, set $\beta = k - 1$, and $\mathcal{G}$.
   7. If $|\mathcal{G}| > t$ and $k = 0$, set $\mathcal{G}$ by Eq. (8).
   8. Update $(H, F, J)$ by Eqs. (7).
9. Set Eq. (4) and compute $d_Y$ by Eqs. (5).
10. while end

Key Idea

Exchange multiple variables among working sets.
- Termination guarantee by a backup rule
- Improved by reduced block exchange

Reduced Block Exchange

Constrain the maximum increase of $|\mathcal{F}|$:

- Allow full exchanges for reducing $|\mathcal{F}|$, i.e., $J_i = J_j$ and $J_j = J_i$.
- Limit the increase of $|\mathcal{F}|$ by enforcing that $|J_i \cup J_j| \leq \alpha p$ where $0 < \alpha < 1$ is a parameter.

Characteristics

- Gaussian structure learning: Block coordinate descent framework of (Banerjee et al., 2008).
- Subproblems are casted as Eq. (2).
- Sparse coding: Will be in longer version.

Extensions

- Iterative optimization schemes (bottom): Reduced Block Exchange
- BPR: Block principal pivoting with full exchange, BPR: BPR with reduced exchange, LARS: (Efron et al., 2004), FS: Feature-search algorithm (Lee et al., 2007), GPR: Gradient projection (Figueiredo et al., 2007), CD: Coordinate descent (Friedman et al., 2008).

Experimental results

Table: Synthetic data sets with sparse random features

<table>
<thead>
<tr>
<th>Problem size</th>
<th>N</th>
<th>k</th>
<th>m</th>
<th>n</th>
<th>p</th>
<th>$\mathcal{G}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500 x 1000</td>
<td>0.31</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>5000 x 2000</td>
<td>0.36</td>
<td>0.37</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>10000 x 5000</td>
<td>0.36</td>
<td>0.37</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Table: Data sets from UCI data repository

| Name          | A | AS | B | MB | PS | Sh | T | UCI | \%
|---------------|---|----|---|----|----|----|---|-----|-----|
| Adult         | 0.3  | 0.25 | 0.25 | 0.3 | 0.25 | 0.25 | 0.3 | 0.25 | 0.3
| Cora          | 0.1  | 0.15 | 0.15 | 0.2 | 0.15 | 0.15 | 0.2 | 0.15 | 0.2
| Iris          | 0.0  | 0.01 | 0.01 | 0.1 | 0.01 | 0.01 | 0.1 | 0.01 | 0.1
| Letter        | 0.0  | 0.01 | 0.01 | 0.1 | 0.01 | 0.01 | 0.1 | 0.01 | 0.1
| MNIST         | 0.0  | 0.01 | 0.01 | 0.1 | 0.01 | 0.01 | 0.1 | 0.01 | 0.1

Figure: (top left): Iteration counts (top right): Execution time (middle left): Speed up by BPR upon (middle right): Iterative optimization schemes (bottom): Gaussian structure learning.

LARS FS BPR

LARS FS BPR

LARS FS BPR

LARS FS BPR

LARS FS BPR