Sparse Linear Programming via Primal and Dual Augmented Coordinate Descent

Presenter: Ian E.H. Yen

Joint work with Kai Zhong, Cho-Jui Hsieh,
Pradeep Ravikumar and Inderjit Dhillon.

Sparse Linear Program

Given vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $m \times n$ matrix

$$A = \begin{bmatrix} A_I \\ A_E \end{bmatrix} = [A_b \; A_f],$$

the *primal* and *dual* forms of Linear Program (LP) are

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ s.t. \quad & A_I x \leq b_I \\ & A_E x = b_E \\ & x_j \geq 0, \;\; j \in [n_b] \end{aligned} \qquad \begin{aligned} \min_{y \in \mathbb{R}^m} \quad & b^T y \\ s.t. \quad & -A_b^T y \leq c_b \\ & -A_f^T y = c_f \\ & y_i \geq 0, \;\; i \in [m_I]. \end{aligned}$$

We say a LP is sparse in the sense that

(i) $nnz(A) \ll m \times n$.
(ii) $nnz(y^*) \ll m$ (dual sparsity).
(iii) $nnz(x^*) \ll n$ (primal sparsity).

Sparse Linear Program: Examples

- L1-regularized Multiclass SVM

$$\begin{aligned} \min_{w_m, \xi_i} \quad & \lambda \sum_{m=1}^{k} \|w_m\|_1 + \sum_{i=1}^{l} \xi_i \\ s.t. \quad & w_{y_i}^T x_i - w_m^T x_i \geq e_i^m - \xi_i, \ \forall (i, m) \end{aligned}$$

- Sparse Inverse Covariance Estimation

$$\begin{aligned} \min_{\Omega \in \mathbb{R}^{d \times d}} \quad & \|\Omega\|_1 \\ s.t. \quad & \|S\Omega - I_d\|_{max} \leq \lambda \end{aligned}$$

- MAP Inference on Factor Graph

$$\begin{aligned} \max_{p_i, i \in V \ q_j, j \in F} \quad & \sum_{j \in F} \theta_j^T q_j \\ s.t. \quad & M_{i,j} q_j = p_i, \ (i,j) \in E \\ & q_j \in \Delta_j. \end{aligned}$$

Algorithms for Linear Program

- **Simplex Method**: Moving between corner points requires solutions of two $m \times m$ linear system ($O(m^3)$ for m iterations). #iterations is exponential in worst case, but between $2m$ and $3m$ typically.
- **Interior Point Method (IPM)**: Reduce LP to series of (asymptotically ill-conditioned) *unconstrained problems* by Barrier functions. Newton Method requires solving an $m \times m$ linear system ($O(m^3)$). #iterations$=O(\log(1/\varepsilon))$ for ε sub-optimality.

Algorithms for Linear Program

- **Simplex Method**: Moving between corner points requires solutions of two $m \times m$ linear system ($O(m^3)$ for m iterations). #iterations is exponential in worst case, but between $2m$ and $3m$ typically.
- **Interior Point Method (IPM)**: Reduce LP to series of (asymptotically ill-conditioned) *unconstrained problems* by Barrier functions. Newton Method requires solving an $m \times m$ linear system ($O(m^3)$). #iterations$=O(\log(1/\varepsilon))$ for ε sub-optimality.
- **Subgradient-based Methods**: Evaluate subgradient takes only $O(nnz(A))$, but requires $O(1/\varepsilon^2)$ iterations. (Even finding a feasible solution is hard.)
- **Augmented Lagrangian Method (ALM):** Reduce LP to series of *bound-constrained Quadratic Problem*. #iterations$=O(\log(1/\varepsilon))$. (cost for solving sub-problem?)

  This paper $\Rightarrow$ Randomized Coordinate Descent (RCD) with ALM gives $O(nnz(A)\log^2(1/\varepsilon))$ overall complexity. More efficient for primal, dual-sparse problems via an Active-Set strategy.

Augmented Lagrangian Method (ALM)

Let $g(y)$ denote the objective of dual LP (taking ∞ for infeasible points). Then a primal ALM is eqivalent to a dual Proximal-Point iterates

$$y^{t+1} = \underset{y}{argmin} \quad g(y) + \frac{1}{2\eta_t}\|y - y^t\|^2, \tag{1}$$

where we find y^{t+1} by solving the dual of (1)

$$\begin{aligned} \min_{x,\,\xi} \quad & c^T x + \frac{\eta_t}{2}\left\| \begin{array}{c} A_I x - b_I + \xi \\ A_E x - b_E \end{array} \right. + \frac{1}{\eta_t}\left[\begin{array}{c} y_I^t \\ y_E^t \end{array} \right]\Bigg\|^2 \\ s.t. \quad & x_b \geq 0,\ \xi \geq 0 \end{aligned}$$

to obtain

$$y^{t+1} = y(x^*, \xi^*) = \eta_t \left[\begin{array}{c} A_I x^* - b_I + \xi^* \\ A_E x^* - b_E \end{array} \right] + \left[\begin{array}{c} y_I^t \\ y_E^t \end{array} \right].$$

ALM with Coordinate Descent (AL-CD)

The quadratic program (with ξ eliminated)

$$\begin{aligned} \min_{x} \quad & F(x) = c^T x + \frac{\eta_t}{2} \left\| \begin{bmatrix} [A_I x - b_I + y_I^t/\eta_t]_+ \\ A_E x - b_E + y_E^t/\eta_t \end{bmatrix} \right\|^2 \\ s.t. \quad & x_b \geq 0. \end{aligned}$$

has

$$\nabla F(x) = c + \eta_t A_I^T [w(x)]_+ + \eta_t A_E^T v(x)$$

where

$$\begin{aligned} w(x) &= A_I x - b_I + y_I^t/\eta_t \\ v(x) &= A_E x - b_E + y_E^t/\eta_t \end{aligned}$$

- Given $w(x)$, $v(x)$, gradient $\nabla_j F(x)$ of coordinate can be evaluated in $O(nnz(a_j))$.
- Maintaining $w(x)$, $v(x)$ after each coordinate j update needs $O(nnz(a_j))$.

ALM with Coordinate Descent (AL-CD)

The quadratic program (with ξ eliminated)

$$\begin{aligned} \min_{x} \quad & F(x) = c^T x + \frac{\eta_t}{2} \left\| \begin{array}{c} [A_I x - b_I + y_I^t/\eta_t]_+ \\ A_E x - b_E + y_E^t/\eta_t \end{array} \right\|^2 \\ s.t. \quad & x_b \geq 0. \end{aligned}$$

RCD algorithm: For each randomly picked $j \in [n]$

1. Solve single-variable QP: [1]

$$d_j^* = \left[x_j - \nabla_j F(x) / \nabla_j^2 F(x) \right]_{j,+} - x_j, \quad \Rightarrow O(nnz(a_j))$$

2. Line search to obtain step size β and $x_j += \beta d_j^*$.
3. Maintain $w(x)$ and $v(x)$. $\Rightarrow O(nnz(a_j))$

[1] $[\,.\,]_{j,+}$ denotes a truncation to 0 if $j \in [n_b]$.

ALM with Coordinate Descent (AL-CD)

The quadratic program (with ξ eliminated)

$$\begin{aligned} \min_{x} \quad & F(x) = c^T x + \frac{\eta_t}{2} \left\| \begin{array}{c} [A_I x - b_I + y_I^t/\eta_t]_+ \\ A_E x - b_E + y_E^t/\eta_t \end{array} \right\|^2 \\ s.t. \quad & x_b \geq 0. \end{aligned}$$

RCD with *Active Set*: For each randomly picked $j \in A^k$

1. Solve single-variable QP: [2]

$$d_j^* = \left[x_j - \nabla_j F(x) / \nabla_j^2 F(x) \right]_{j,+} - x_j,$$

2. Line search to obtain step size β and $x_j += \beta d_j^*$.
3. Maintain $w(x)$ and $v(x)$.
4. Remove coordinate j from A^k if $\nabla_j F(x) > \varepsilon_A$ and $j \in [n_b]$.

[2] $[\,.\,]_{j,+}$ denotes a truncation to 0 if $j \in [n_b]$.

Convergence Analysis

- Subprobleem is strongly convex when restricted to $N(\bar{A})^{\perp}$ (CNSC) $\Rightarrow$ We show RCD converges to $F(x) - F(x^*) \leq \varepsilon$ in $\gamma n \log(1/\varepsilon)$ number of iterations w.h.p., and the cost for sub-problem is $O(nnz(A)\log(1/\varepsilon))$
- The ALM does not amplify error when solving subproblem inexactly. $\Rightarrow$ If each subproblem is approximated to ε then after t iterations we have approximation error at most $t\varepsilon$.
- Needs $O(tn\log(t/\varepsilon)) = O(n\log^2(1/\varepsilon))$ CD iterations overall.

Implementation Details

- Two-Phase strategy:
  Phase-I: Spend constant cost on each sub-problem. ($|A|$ is large)
  Phase-II: Solve each sub-problem to precision ε_t, with η_t, ε_t dynamically adjusted. ($|A|$ is small)
- For ill-conditioned problem, use Projected-Newton-CG when Active set becomes stable.

Experiments

Table 1: Timing Results (in sec. unless specified o.w.) on Multiclass L1-regularized SVM

Data	n_b	m_I	P-Simp.	D-Simp.	Barrier	D-ALCD	P-ALCD
rcv1	4,833,738	778,200	$>$ 48hr	$>$ 48hr	$>$ 48hr	3,452	**3,155**
news	2,498,415	302,765	$>$ 48hr	37,912	$>$ 48hr	**148**	395
sector	11,597,992	666,848	$>$ 48hr	9,282	$>$ 48hr	**1,419**	2,029
mnist	75,620	540,000	6,454	2,556	73,036	**146**	7,207
cod-rna.rf	69,537	59,535	86,130	5,738	$>$ 48hr	3,130	**2,676**
vehicle	79,429	157,646	3,296	143.33	8,858	**31**	598
real-sim	114,227	72,309	$>$ 48hr	49,405	89,476	**179**	297

L1-regularized Multiclass SVM: $nnz(A) \ll mn$, $nnz(y^*) \ll m$, and $nnz(x^*) \ll n$.

Experiments

Table 2: Timing Results (in sec. unless specified o.w.) on Sparse Inverse Covariance Estimation

Data	n_b	m_I	m_E	n_f	P-Simp	D-Simp	Barrier	D-ALCD	P-ALCD
textmine	60,876	60,876	43,038	43,038	$>$ 48hr	$>$ 48hr	$>$ 48hr	43,096	**18,507**
E2006	55,834	55,834	32,174	32,174	$>$ 48hr	$>$ 48hr	94623	$>$ 48hr	**4,207**
dorothea	47,232	47,232	1,600	1,600	3,980	103	82	47	**38**

- Sparse Inverse Covariance Estimation:

$$\begin{aligned} \min_{\Omega \in \mathbb{R}^{d \times d}} \quad & \|\Omega\|_1 \\ s.t. \quad & \|S\Omega - I_d\|_{max} \leq \lambda \end{aligned}$$

- $S = Z^T Z$, Z is $n \times d \ll d^2 \Rightarrow$ Transform to:

$$\begin{aligned} \min_{\Omega \in \mathbb{R}^{d \times d}, Y \in \mathbb{R}^{n \times d}} \quad & \|\Omega\|_1 \\ s.t. \quad & \|Z^T Y - I_d\|_{max} \leq \lambda \\ & Y = Z\Omega \end{aligned}$$

- $nnz(A) \ll mn$ and $nnz(x^*) \ll n$.

Experiments

Table 3: Timing Results (in sec. unless specified o.w.) for Nonnegative Matrix Factorization.

Data	n_b	m_I	P-Simp.	D-Simp.	Barrier	D-ALCD	P-ALCD
micromass	2,896,770	4,107,438	$>$ 96hr	$>$ 96hr	280,230	12,966	**12,119**
ocr	6,639,433	13,262,864	$>$ 96hr	$>$ 96hr	284,530	**40,242**	$>$ 96hr

NMF: $nnz(A) \ll mn$.

Future Works

- Utilize Active-set in a non-heuristic way. Can we not go through all variables each ALM iteration?
- Exploit primal, dual sparsity simultaneously.