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

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

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

$$\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{s.t.} & \quad A_l x \leq b_l \\
& \quad A_E x = b_E \\
& \quad x_j \geq 0, \ j \in [n_b]
\end{align*}$$

$$\begin{align*}
\min_{y \in \mathbb{R}^m} & \quad b^T y \\
\text{s.t.} & \quad -A_b^T y \leq c_b \\
& \quad -A_f^T y = c_f \\
& \quad y_i \geq 0, \ i \in [m_l].
\end{align*}$$

We say a LP is sparse in the sense that

(i) $\text{nnz}(A) \ll m \times n$.
(ii) $\text{nnz}(y^*) \ll m$ (dual sparsity).
(iii) $\text{nnz}(x^*) \ll n$ (primal sparsity).
Sparse Linear Program: Examples

- **L1-regularized Multiclass SVM**

  $$\min_{w_m, \xi_i} \lambda \sum_{m=1}^{k} \|w_m\|_1 + \sum_{i=1}^{l} \xi_i$$

  $$s.t. \quad w_{y_i}^T x_i - w_m^T x_i \geq e^*_i - \xi_i, \forall (i, m)$$

- **Sparse Inverse Covariance Estimation**

  $$\min_{\Omega \in \mathbb{R}^{d \times d}} \|\Omega\|_1$$

  $$s.t. \quad \|S\Omega - I_d\|_{max} \leq \lambda$$

- **MAP Inference on Factor Graph**

  $$\max_{p_i, i \in V, q_j, j \in F} \sum_{j \in F} \theta_j^T q_j$$

  $$s.t. \quad M_{i,j} q_j = p_i, (i, j) \in E$$

  $$q_j \in \Delta_j.$$
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 $\varepsilon$ 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 $\varepsilon$ 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 \( \infty \) for infeasible points). Then a primal ALM is equivalent to a dual Proximal-Point iterates

\[
y^{t+1} = \underset{y}{\text{argmin}} \; g(y) + \frac{1}{2\eta_t} \|y - y^t\|^2,
\]

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

\[
\min_{x, \xi} \; c^T x + \frac{\eta_t}{2} \left\| \begin{bmatrix} A_I x - b_I + \xi \\ A_E x - b_E \end{bmatrix} \right\|^2
\]

s.t. \( x_b \geq 0, \; \xi \geq 0 \)

to obtain

\[
y^{t+1} = y(x^*, \xi^*) = \eta_t \begin{bmatrix} A_I x^* - b_I + \xi^* \\ A_E x^* - b_E \end{bmatrix} + \begin{bmatrix} y^t_I \\ y^t_E \end{bmatrix}.
\]
ALM with Coordinate Descent (AL-CD)

The quadratic program (with $\xi$ eliminated)

$$
\min_{x} \quad F(x) = c^T x + \frac{\eta_t}{2} \left\| \begin{bmatrix} A_I x - b_I + y_I^t/\eta_t \\ A_E x - b_E + y_E^t/\eta_t \end{bmatrix}^+ \right\|_2^2
$$

s.t. $x_b \geq 0$.

has

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

where

$$
w(x) = A_I x - b_I + y_I^t/\eta_t
$$

$$
v(x) = A_E x - b_E + y_E^t/\eta_t
$$

- 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 $\xi$ eliminated)

$$
\min_x F(x) = c^T x + \frac{\eta_t}{2} \left\| \begin{array}{c}
A_l x - b_l + y^t_l / \eta_t \\
A_E x - b_E + y^t_E / \eta_t
\end{array} \right\|_2^2
$$

s.t. $x_b \geq 0$.

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

1 Solve single-variable QP: \footnote{\([.]_{j,+}\) denotes a truncation to 0 if $j \in [n_b]$.}

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

2 Line search to obtain step size $\beta$ and $x_j^+ = \beta d_j^*$.

3 Maintain $w(x)$ and $v(x). \Rightarrow O(nnz(a_j))$
ALM with Coordinate Descent (AL-CD)

The quadratic program (with $\xi$ eliminated)

$$\min_x F(x) = c^T x + \frac{\eta t}{2} \left\| \begin{bmatrix} A_I x - b_I + y_t^I / \eta_t \\ A_E x - b_E + y_t^E / \eta_t \end{bmatrix} + \begin{bmatrix} x_b \geq 0 \end{bmatrix} \right\|^2$$

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 $\beta$ 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 $j \in [n_b]$. 

Ian E.H. Yen
Convergence Analysis

- Subproblem 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 $\varepsilon$ then after $t$ iterations we have approximation error at most $t\varepsilon$.

- Needs $O(t n \log(t/\varepsilon)) = O(n \log^2(1/\varepsilon))$ 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 $\varepsilon_t$, with $\eta_t$, $\varepsilon_t$ 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

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

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

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

<table>
<thead>
<tr>
<th>Data</th>
<th>$n_b$</th>
<th>$m_I$</th>
<th>$m_E$</th>
<th>$n_f$</th>
<th>P-Simp</th>
<th>D-Simp</th>
<th>Barrier</th>
<th>D-ALCD</th>
<th>P-ALCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>textmine</td>
<td>60,876</td>
<td>60,876</td>
<td>43,038</td>
<td>43,038</td>
<td>&gt; 48hr</td>
<td>&gt; 48hr</td>
<td>&gt; 48hr</td>
<td>43,096</td>
<td>18,507</td>
</tr>
<tr>
<td>E2006</td>
<td>55,834</td>
<td>55,834</td>
<td>32,174</td>
<td>32,174</td>
<td>&gt; 48hr</td>
<td>&gt; 48hr</td>
<td>94623</td>
<td>&gt; 48hr</td>
<td>4,207</td>
</tr>
<tr>
<td>dorothea</td>
<td>47,232</td>
<td>47,232</td>
<td>1,600</td>
<td>1,600</td>
<td>3,980</td>
<td>103</td>
<td>82</td>
<td>47</td>
<td>38</td>
</tr>
</tbody>
</table>

- Sparse Inverse Covariance Estimation:
  \[
  \min_{\Omega \in \mathbb{R}^{d \times d}} \| \Omega \|_1 \\
  \text{s.t.} \quad \| S\Omega - I_d \|_{\text{max}} \leq \lambda
  \]

  - $S = Z^T Z$, $Z$ is $n \times d \ll d^2 \Rightarrow$ Transform to:
    \[
    \min_{\Omega \in \mathbb{R}^{d \times d}, Y \in \mathbb{R}^{n \times d}} \| \Omega \|_1 \\
    \text{s.t.} \quad \| Z^T Y - I_d \|_{\text{max}} \leq \lambda \\
    Y = Z\Omega
    \]

- $\text{nnz}(A) \ll mn$ and $\text{nnz}(x^*) \ll n$. 
Experiments

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

<table>
<thead>
<tr>
<th>Data</th>
<th>$n_b$</th>
<th>$m_I$</th>
<th>P-Simp.</th>
<th>D-Simp.</th>
<th>Barrier</th>
<th>D-ALCD</th>
<th>P-ALCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>micromass</td>
<td>2,896,770</td>
<td>4,107,438</td>
<td>&gt; 96hr</td>
<td>&gt; 96hr</td>
<td>280,230</td>
<td>12,966</td>
<td>12,119</td>
</tr>
<tr>
<td>ocr</td>
<td>6,639,433</td>
<td>13,262,864</td>
<td>&gt; 96hr</td>
<td>&gt; 96hr</td>
<td>284,530</td>
<td>40,242</td>
<td></td>
</tr>
</tbody>
</table>

NMF: $\text{nnz}(A) \ll mn$. 
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.