A New Alternating Direction Method for Linear Programming

Sinong Wang Ness Shroff
The Ohio State University

Research Problems

We are interested in designing the fast algorithm to solve the general **linear programing (LP)** problem of the form,

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad \mathbf{c}^T\mathbf{x}$$
$$s.t. \quad \mathbf{A}\mathbf{x}=\mathbf{b}, x_i \geq 0, i\in[n_b]$$

Applications of LP in machine learning:

- l1-regularized support vector machine (SVM) problem.
- Nonnegative matrix factorization problem.
- Sparse inverse covariance matrix estimation problem.
- Markov decision process (MDP) problem.
- Maximum a posterior estimation problem

Basic features of LP in machine learning: large-scale, sparse

$$nnz(\mathbf{A}) \ll mn$$

Existing Algorithms including simplex method and interior point method: complexity is at least quadratic in the problem dimension.

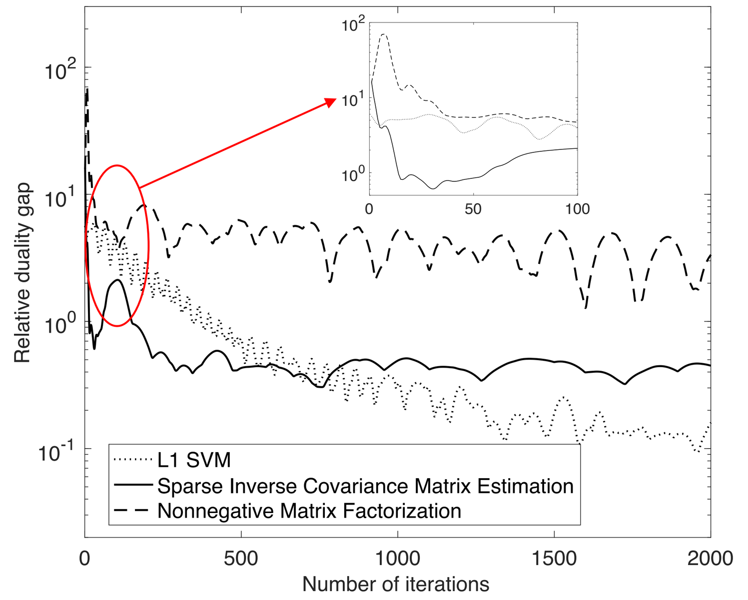
Research objective: design an algorithm to exploit the sparse structure.

Related Works

First-order algorithm requires a matrix vector multiplication $\mathbf{Ax}$ in each iteration with complexity linear in nnz(A).

- Subgradient descent method
- Augmented Lagrangian Method (ALM) [2]
- **Alternating Directional Method of Multiplier (ADMM) [1]**

Tail Convergence of the Existing ADMM Method [1]



Observations:

- It converges fast in the initial phase, but exhibits a slow and fluctuating tail convergence.
- Theoretically, it can be recovered by an inexact Uzawa method (local second-order approximates augmented Lagrangian function)

New Variable Splitting Method

We separate the equality and inequality constraints by adding another group of variables y.

Primal	Dual
$\min \quad \mathbf{c}^T\mathbf{x}$ $s.t. \quad \mathbf{Ax}=\mathbf{b}, \mathbf{x}=\mathbf{y}$ $y_i \geq 0, i\in[n_b].$	$\min \quad \mathbf{b}^T\mathbf{z}_x$ $s.t. \quad -\mathbf{A}^T\mathbf{z}_x - \mathbf{z}_y = \mathbf{c},$ $z_{y,i} \leq 0, i\in[n_b],$ $z_{y,i} = 0, i\in[n]\backslash[n_b].$

The Augmented Lagrangian function of the primal problem is

$$L(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathbf{c}^T\mathbf{x} + g(\mathbf{y}) + \mathbf{z}^T(\mathbf{A}_1\mathbf{x}+\mathbf{A}_2\mathbf{y}-\overline{\mathbf{b}}) + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}+\mathbf{A}_2\mathbf{y}-\overline{\mathbf{b}}\|^2$$

Gauss–Seidel type update:

- Primal: $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} L(\mathbf{x},\mathbf{y}^k,\mathbf{z}^k)$ $\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}\in\mathbb{R}^n} L(\mathbf{x}^{k+1},\mathbf{y},\mathbf{z}^k)$
- Dual update: $\mathbf{z}^{k+1} = \mathbf{z}^k + \rho(\mathbf{A}_1\mathbf{x}^{k+1} + \mathbf{A}_2\mathbf{y}^{k+1} - \overline{\mathbf{b}})$

Basic **feature** of this algorithm:

- Update of x can be reduced to solving a well-conditioned linear system
$$\mathbf{x}^{k+1} = \rho^{-1}(\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{d}^k$$
- Update of y can be solved in closed-form expression.

Global Linear Convergence of New Splitting Method

Lemma 1 (**Convergence** [3]). Let $\mathbf{p}^k = \mathbf{z}^k - \rho\mathbf{A}_2\mathbf{y}^k$, then

$$\|\mathbf{p}^{k+1} - [\mathbf{p}^{k+1}]_{G^*}\|^2 \leq \|\mathbf{p}^k - [\mathbf{p}^k]_{G^*}\|^2 - \|\mathbf{p}^{k+1} - \mathbf{p}^k\|^2$$

Distance to optimal solution set monotonically **decreases**.

Lemma 2 (**Geometry of the optimal solution set of LP**)

- Feasibility: $\mathbf{Ax}^* = \mathbf{b}, \mathbf{x}^* = \mathbf{y}^*$ and $-\mathbf{A}^T\mathbf{z}_{\mathbf{x}}^* - \mathbf{z}_y^* = \mathbf{c}$
$$y_i^* \geq 0, z_{y,i}^* \leq 0, i\in[n_b]; \ z_{y,i}^* = 0, i\in[n]\backslash[n_b]$$
- Strong duality: $\mathbf{c}^T\mathbf{x}^* + \mathbf{b}^T\mathbf{z}_x^* = 0.$

Optimal set of LP is described by a **convex polyhedron.**

Lemma 3 (**Hoffman bound [4]**) $\mathcal{S} = \{\mathbf{x}\in\mathbb{R}^d | \mathbf{Ex}=\mathbf{t}, \mathbf{Cx}\leq\mathbf{d}\}$

$$\|\mathbf{x} - [\mathbf{x}]_{\mathcal{S}}\|^2 \leq \theta_{\mathcal{S}}(\|\mathbf{Ex}-\mathbf{t}\|^2 + \|[\mathbf{Cx}-\mathbf{d}]_+\|^2$$

Bound distance by residuals (constraint violations).

Lemma 4 (**Estimation of residuals**)

$$\begin{cases} \mathbf{A}_1\mathbf{x}^{k+1} + \mathbf{A}_2\mathbf{y}^k - \overline{\mathbf{b}} = (\mathbf{p}^{k+1} - \mathbf{p}^k)/\rho, \\ \mathbf{c} + \mathbf{A}_1^T\mathbf{z}^k = \mathbf{A}_1^T(\mathbf{p}^k - \mathbf{p}^{k+1}), \\ \mathbf{c}^T\mathbf{x}^{k+1} + \mathbf{b}^T\mathbf{z}_x^k = (\mathbf{A}_1\mathbf{x}^{k+1} - \mathbf{z}^k/\rho)^T(\mathbf{p}^k - \mathbf{p}^{k+1}), \\ y_i^k \geq 0, z_{y,i}^k \leq 0, i\in[n_b]; z_{y,i}^k = 0, i\in[n]\backslash[n_b]. \end{cases}$$

Theorem 1 (**Global linear convergence**) To guarantee that $\|\mathbf{z}^k - \mathbf{z}^*\| \leq \epsilon$, it suffices to run $K = 2\gamma^2\log(2D_0/\epsilon)$ ADMM iterations with solving accuracy $\epsilon_k \leq \epsilon^2/8K^2$.

Theorem 2 (**Overall Complexity**) If we use the ACDM[5] to solve the inner linear system, the overall complexity of algorithm 1 is $O(a_m\theta_{S^*}^2(R_x\|\mathbf{A}\| + R_z)^2 nnz(\mathbf{A})\log^2(1/\epsilon))$

The complexity of existing ADMM [1] is

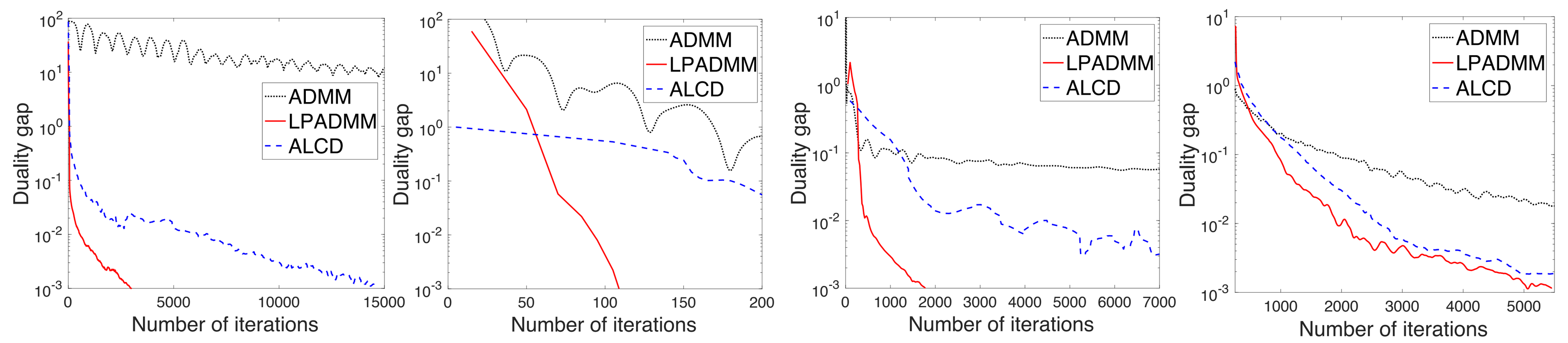
$$O(a_m\mu^2(a_m R_x + d_m R_z)^2(\sqrt{mn} + \|\mathbf{A}\|_F)^2 nnz(\mathbf{A})\log(1/\epsilon)).$$

Algorithm 1 Alternating Direction Method of Multiplier with Inexact Subproblem Solver

Initialize $\mathbf{z}^0 \in \mathbb{R}^{m+n}$, choose parameter $\rho > 0$.
repeat
1. Primal update: update x by solving the linear system with accuracy ϵ_k,
$\mathbf{x}^{k+1} = \rho^{-1}(\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{d}^k$, with $\mathbf{d}^k = -\mathbf{A}_1^T[\mathbf{z}^k + \rho(\mathbf{A}_2\mathbf{y}^k - \overline{\mathbf{b}})] - \mathbf{c}$.
2. Primal update: for each i, let $y_i^{k+1} = [x_i^{k+1} + z_{y,i}^k/\rho]_{V_i}$.
3. Dual update: $\mathbf{z}_x^{k+1} = \mathbf{z}_x^k + \rho(\mathbf{Ax}^{k+1} - \mathbf{b})$, $\mathbf{z}_y^{k+1} = \mathbf{z}_y^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$.

until $\|\mathbf{Ax}^{k+1} - \mathbf{b}\|_\infty \leq \epsilon$ and $\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|_\infty \leq \epsilon$

Simulation Results



Timing Results (in sec. long means > 60 hours)

Data	m	n	nnz($\mathbf{A}$)	LPADMM		ALCD		ADMM	
				Time	Iterations	Time	Iterations	Time	Iterations
bp1	17408	16384	8421376	**22**	**3155**	864	14534	long	long
bp2	34816	32768	33619968	**79**	**4657**	2846	19036	long	long
bp3	69632	65536	134348800	**217**	**6287**	12862	24760	long	long
arcene	50095	30097	1151775	**801**	**15198**	1978	176060	21329	2035415
real-sim	176986	135072	7609186	**955**	**4274**	1906	18262	19697	249363
sonar	80912	68224	2756832	**258**	**5446**	659	13789	3828	151972
colon	217580	161040	8439626	**395**	**216**	455	1288	7423	83680
w2a	12048256	12146960	167299110	**19630**	**2525**	45388	8492	long	long
news20	2785205	2498375	53625267	**7765**	**2205**	9173	6174	long	long

- **2X-40X** speed up compared with state-of-arts.
- **Significantly faster** than commercial software CPLEX
- **Flexibility** to tackle various problems.

[1] Jonathan Eckstein, Dimitri P Bertsekas. An Alternating Direction Method for Linear Programming, 1990.
[2] Ian En-Hsu Yen, et al. Sparse Linear Programming via Primal and Dual Augmented Coordinate Descent. *NIPS* 2015.
[3] Eckstein, Jonathan, Dimitri P. Bertsekas. "On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators." *Mathematical Programming*, 1992.
[4] Alan J Hoffman. On Approximate Solutions of Systems of Linear Inequalities. Journal of Research of the National Bureau of Standards , 1952.
[5] Zeyuan Allen-Zhu, et al. Even Faster Accelerated Coordinate Descent Using Non-uniform Sampling. In *ICML* 2016.