Solving Stackelberg Prediction Game with Least Squares Loss via Spherically Constrained Least Squares Reformulation

JIANG, Rujun

School of Data Science
Fudan University

Fudan University June 25, 2022

(Joint work with Jiali Wang, Wen Huang, Xudong Li and Alex L. Wang)

Introduction

- adversarial learning: the data provider attempts to fool models by supplying deceptive input
- stackelberg prediction game (SPG)[1]: model the interaction between learner (leader) and data generators (follower)
- applications: intrusion detection, banking fraud detection, spam filtering, malware detection, and cybersecurity adversarial attacks
  - email spam senders design message templates that are instantiated by nodes of botnets; templates are specifically designed to produce a low spam score with current spam filters.
  - a fraudster's goal: maximize the profit made from exploiting phished account information
  - an email service provider's goal: achieve a high spam recognition rate at close-to-zero false positives

---

[1] Brückner, M. and Scheffer, T. Stackelberg games for adversarial prediction problems. KDD 2011.

SPG-LS Problem Statement

given a sample $S = \{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^m$

- $\mathbf{x}_i \in \mathbb{R}^n$, the input example
- y_i, the output labels of interests to the learner
- z_i, the output labels of interests to the data provider

setting: the learner aims to train a linear predictor $\mathbf{w}$ based on S; being aware of the learner's predictor $\mathbf{w}$, adversarial data provider intends to fool the learner to predict the label z by modifying the input data $\mathbf{x}$ to $\widehat{\mathbf{x}}$

- the cost of data provider:

$$\mathbf{x}_i^* = \underset{\widehat{\mathbf{x}}_\mathbf{i}}{\operatorname{argmin}} \ \|\mathbf{w}^T\widehat{\mathbf{x}}_\mathbf{i} - z_i\|^2 + \gamma\|\widehat{\mathbf{x}}_\mathbf{i} - \mathbf{x}_\mathbf{i}\|_2^2 \quad i \in [m].$$

- the cost of the learner:

$$\mathbf{w}^* \in \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^m \|\mathbf{w}^T\mathbf{x}_i^* - y_i\|^2.$$

Bilevel and QFP Formulations

setting $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}^T$ and $\widehat{X} = \{\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2, ..., \widehat{\mathbf{x}}_m\}^T$ gives

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|X^*\mathbf{w} - \mathbf{y}\|^2 \\ \text{s.t.} \quad & X^* = \operatorname*{argmin}_{\widehat{X}} \|\widehat{X}\mathbf{w} - \mathbf{z}\|^2 + \gamma\|\widehat{X} - X\|_F^2 \end{aligned} \tag{1}$$

- optimality condition of lower level problem

$$X^* = \left(\mathbf{z}\mathbf{w}^T + \gamma X\right)\left(\mathbf{w}\mathbf{w}^T + \gamma I\right)^{-1}$$

- use Sherman-Morrison formula

$$X^*\mathbf{w} = \frac{\frac{1}{\gamma}\mathbf{z}\mathbf{w}^T\mathbf{w} + X\mathbf{w}}{1 + \frac{1}{\gamma}\mathbf{w}^T\mathbf{w}}$$

reformulate the bilevel problem into quadratic fractional program (QFP)

$$\begin{aligned} \min_{\mathbf{w},\alpha} \quad & \frac{\|\frac{\alpha}{\gamma}\mathbf{z} + X\mathbf{w} - \mathbf{y} - \frac{\alpha}{\gamma}\mathbf{y}\|^2}{(1 + \frac{\alpha}{\gamma})^2} \\ \text{s.t.} \quad & \alpha = \mathbf{w}^T\mathbf{w} \end{aligned} \tag{2}$$

Literature Review: Bisection Method

the celebrated Dinkelbach's theorem[2]: $(\mathbf{w}, \alpha)$ is a solution of (2) if and only if the optimal value of (3) is 0

$$\begin{aligned} F(q) := \min_{\mathbf{w},\alpha} \quad & \|\frac{\alpha}{\gamma}\mathbf{z} + X\mathbf{w} - \mathbf{y} - \frac{\alpha}{\gamma}\mathbf{y}\|^2 - q(1+\frac{\alpha}{\gamma})^2 \\ \text{s.t} \quad & \alpha = \mathbf{w}^T\mathbf{w}. \end{aligned} \tag{3}$$

bisection method[3]:

- apply a bisection search for q^* such that $F(q^*) = 0$
- each inner subproblem solves (3)
- initial lower and upper bound can be constructed from data

[2]Dinkelbach,W. On nonlinear fractional programming. Management Science, 13(7):492–498, 1967.

[3]Bishop, N., Tran-Thanh, L., and Gerding, E. Optimal learning from verified training data. In Advances in Neural Information Processing Systems 33, 2020.

Literature Review: SDP Method[4]

main problem: the following equivalent formulation of SPG-LS

$$\min_{\mathbf{w},\ \alpha} \quad \frac{\|\alpha\mathbf{z} + X\mathbf{w} - \mathbf{y} - \alpha\mathbf{y}\|^2}{(1+\alpha)^2} \qquad \text{s.t.} \quad \frac{\mathbf{w}^T\mathbf{w}}{\gamma} = \alpha \tag{4}$$

define

$$A = \begin{pmatrix} X^TX & X^T(\mathbf{z}-\mathbf{y}) & -X^T\mathbf{y} \\ (\mathbf{z}-\mathbf{y})^TX & \|\mathbf{z}-\mathbf{y}\|^2 & -(\mathbf{z}-\mathbf{y})^T\mathbf{y} \\ -\mathbf{y}^TX & -\mathbf{y}^T(\mathbf{z}-\mathbf{y}) & \mathbf{y}^T\mathbf{y} \end{pmatrix},$$

$$B = \begin{pmatrix} \mathbf{0}_n & & \\ & 1 & 1 \\ & 1 & 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} \frac{I_n}{\gamma} & & \\ & 0 & -\frac{1}{2} \\ & -\frac{1}{2} & 0 \end{pmatrix}$$

problem (4) is equivalent to the following semidefinite program (SDP)

$$\begin{array}{ll} \sup_{\mu,\lambda} & \mu \\ \text{s.t.} & A - \mu B + \lambda C \succeq 0 \end{array} \tag{5}$$

[4] Wang, J., Chen, H., Jiang, R., Li, X., and Li, Z. Fast algorithms for stackelberg prediction game with least squares loss. ICML 2021.

Literature Review: SOCP Method[5]

A, B and C are simultaneously congruent to arrow matrices, i.e.,

$$\tilde{A} = V^T A V = \begin{pmatrix} D & \mathbf{b} \\ \mathbf{b}^T & c \end{pmatrix}, \tilde{B} = VBV = \begin{pmatrix} \mathcal{O}_{n+1} & \\ & 4 \end{pmatrix}, \tilde{C} = V^T C V = \begin{pmatrix} \frac{1}{\gamma} I_{n+1} & \\ & -1 \end{pmatrix}.$$

where $D = \mathrm{Diag}(d_1, \ldots, d_{n+1}) \in \mathbb{R}^{(n+1)\times(n+1)}$

then LMI $A - \mu B + \lambda C \succeq 0$ is equivalent to

$$\tilde{A} - \mu \tilde{B} + \lambda \tilde{C} \succeq 0, \tag{6}$$

(6) is further equivalent to the second order cone program (SOCP):

$$\begin{array}{ll} \sup_{\mu,\lambda,\mathbf{s}} & \mu \\ \text{s.t.} & d_i + \frac{\lambda}{\gamma} \geq 0,\ i \in [n+1], \\ & c - 4\mu - \lambda - \sum_{i=1}^{n+1} s_i \geq 0, \\ & s_i(d_i + \frac{\lambda}{\gamma}) \geq\ b_i^2,\ s_i \geq 0\ i \in [n+1], \end{array} \tag{7}$$

[5]Wang, J., Chen, H., Jiang, R., Li, X., and Li, Z. Fast algorithms for stackelberg prediction game with least squares loss. ICML 2021.

Existing Methods

- Speed: SOCP $\gg$ SDP $\gg$ Bisection
- The SOCP is still not well-suited for solving large-scale SPG-LS, although it is much faster than the SDP approach. The spectral decomposition is time-consuming.
- This paper proposed factorization-free methods for the practical applicability of SPG-LS.

SCLS Reformulation (Main Result)

quadratic fractional programming (QFP) reformulation of SPG-LS

$$\begin{array}{ll} \min\limits_{\mathbf{w},\ \alpha} & v(\mathbf{w},\alpha) \triangleq \dfrac{\left\|\alpha\mathbf{z} + X\mathbf{w} - \mathbf{y} - \alpha\mathbf{y}\right\|^2}{(1+\alpha)^2} \\ \text{s.t.} & \frac{\mathbf{w}^T\mathbf{w}}{\gamma} = \alpha \end{array} \tag{8}$$

define

$$\tilde{\mathbf{w}} := \frac{2}{\sqrt{\gamma}(\alpha+1)}\mathbf{w} \quad \text{and} \quad \tilde{\alpha} := \frac{\alpha-1}{\alpha+1},$$

then (8) is equivalent to the Spherically Constrained Least Squares (SCLS)

$$\begin{array}{ll} \min\limits_{\tilde{\mathbf{w}},\, \tilde{\alpha}} & \tilde{v}(\tilde{\mathbf{w}},\tilde{\alpha}) \triangleq \left\| \frac{\tilde{\alpha}}{2}\mathbf{z} + \frac{\sqrt{\gamma}}{2}X\tilde{\mathbf{w}} - (\mathbf{y} - \frac{\mathbf{z}}{2}) \right\|^2 \\ \text{s.t.} & \tilde{\mathbf{w}}^T\tilde{\mathbf{w}} + \tilde{\alpha}^2 = 1. \end{array} \tag{9}$$

Let $\mathbf{r} = \left(\begin{smallmatrix} \tilde{\mathbf{w}} \\ \tilde{\alpha} \end{smallmatrix} \right)$, we have compact form

$$\min_{\mathbf{r}} \; q(\mathbf{r}) \quad \text{s.t. } \mathbf{r}^T \mathbf{r} = 1,$$

where $q(\mathbf{r})$ is a least squares

$$q(\mathbf{r}) = \|\widehat{L}\mathbf{r} - (\mathbf{y} - \mathbf{z}/2)\|_2^2 = \mathbf{r}^T H \mathbf{r} + 2\mathbf{g}^T \mathbf{r} + p$$

Sketch of Proof

given a feasible solution in (8), construct a feasible solution with the same objective value in (9) and vice versa

- Suppose $(\mathbf{w}, \alpha)$ is a feasible solution of (8). Then $(\tilde{\mathbf{w}}, \tilde{\alpha})$, defined as

  $$\tilde{\mathbf{w}} := \frac{2}{\sqrt{\gamma}(\alpha+1)}\mathbf{w} \quad \text{and} \quad \tilde{\alpha} := \frac{\alpha-1}{\alpha+1},$$

  is feasible to (9) and $v(\mathbf{w}, \alpha) = \tilde{v}(\tilde{\mathbf{w}}, \tilde{\alpha})$.

- Suppose $(\tilde{\mathbf{w}}, \tilde{\alpha})$ is feasible to (8) with $\tilde{\alpha} \neq 1$. Then $(\mathbf{w}, \alpha)$, defined as

  $$\mathbf{w} := \frac{\sqrt{\gamma}}{1-\tilde{\alpha}}\tilde{\mathbf{w}} \quad \text{and} \quad \alpha := \frac{1+\tilde{\alpha}}{1-\tilde{\alpha}},$$

  is feasible to (9) and $\tilde{v}(\tilde{\mathbf{w}}, \tilde{\alpha}) = v(\mathbf{w}, \alpha)$.

Practical Algorithms for SCLS

❶ Krylov Subspace Method

- $(k+1)$st Krylov subspace: $\mathcal{K}_k := \{\mathbf{g}, H\mathbf{g}, H^2\mathbf{g}, \ldots, H^k\mathbf{g}\}$
- $Q_k = [\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_k] \in \mathbb{R}^{(n+1)\times(k+1)}$: an orthonormal basis produced by the generalized Lanczos process
- subproblem in $(k+1)$st Krylov subspace:

$$\min_{r\in\mathcal{K}_k, \|\mathbf{r}\|=1} \mathbf{r}^T H\mathbf{r} + 2\mathbf{g}^T\mathbf{r} + p.$$

Convergence rate [6]

Lagrangian multiplier λ; $\kappa = \frac{\lambda_{\max}+\lambda^*}{\lambda_{\min}+\lambda^*}$ condition number of the SCLS

- If $\kappa < \infty$: $f(\mathbf{r}_k) - f(\mathbf{r}^*) \leq \mathcal{O}(\exp(-k/\sqrt{\kappa}))$
- If $\mathbf{g}$ is perturbed with random vector: $f(\mathbf{r}_k) - f(\mathbf{r}^*) \leq \tilde{\mathcal{O}}(1/k^2)$

In summary, $\tilde{\mathcal{O}}(N/\sqrt{\epsilon})$ time algorithms for an ϵ optimal solution

[6] Carmon, Y. and Duchi, J. C. Analysis of krylov subspace solutions of regularized nonconvex quadratic problems. In *NeurIPS*, pp. 10728–10738, 2018.

❷ Riemannian Trust Region Newton (RTRNewton) Method[7]

- manifold: unit sphere $\mathcal{S}^n = \{\mathbf{r} \in \mathbb{R}^{n+1} : \mathbf{r}^T\mathbf{r} = 1\}$
- similar to standard trust region Newton method
- both Riemannian gradient and Riemannian hessian vector product can be obtained in $\mathcal{O}(n^2)$ or $\mathcal{O}(N)$

Convergence rate

- local convergence: $\mathrm{grad}\ q(\mathbf{r}^*) = 0$, $\mathrm{Hess}\ q(\mathbf{r}^*) \succeq 0$

$$\mathrm{dist}(\mathbf{r}_{k+1}, \mathbf{r}^*) \leq c\ \mathrm{dist}(\mathbf{r}_k, \mathbf{r}^*)^2$$

- global convergence: $\|\mathrm{grad}(q(x))\| \leq \epsilon_g$, iteration complexity $\mathcal{O}\left(\epsilon_g^{-2}\right)$

[7]Absil, Baker, and Gallivan] Absil, P.-A., Baker, C. G., and Gallivan, K. A. Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

Complexity Comparison

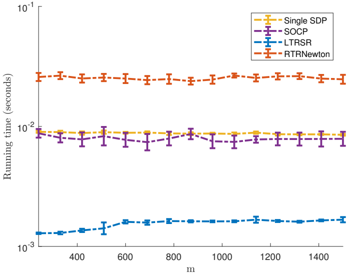
RTRNewton cannot guarantee global minimum, iteration complexity $\mathcal{O}\left(\epsilon_g^{-2}\right)$

1. Krylov subspace method
   - sparse case: $\mathcal{O}(N \log(1/\epsilon))$ if $\kappa < \infty$;
     in general $\tilde{\mathcal{O}}(N/\sqrt{\epsilon})$
   - dense case: $\mathcal{O}(n^2 \log(1/\epsilon))$ if assume $m = \mathcal{O}(n)$ and $\kappa < \infty$;
     in general $\tilde{\mathcal{O}}(n^2/\sqrt{\epsilon})$
2. SOCP method:$\mathcal{O}\left(n^w + n^{\frac{3}{2}} \log(1/\epsilon)\right), 2 < w < 3$

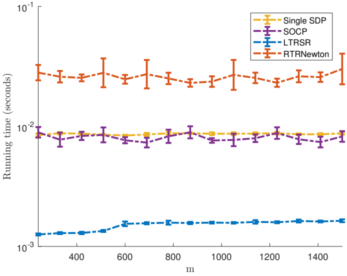
   formulate matrix A + spectral decomposition + IPM for SOCP

   however, no implementable algorithm for spectral decomposition with complexity $\mathcal{O}(n^w)$

Experiments: Real Datasets

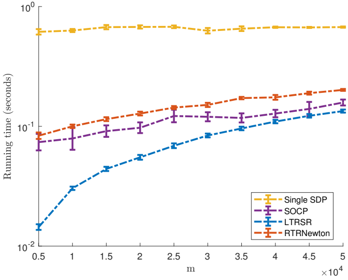


(a) wine data: $\mathcal{A}_{\text{modest}}$

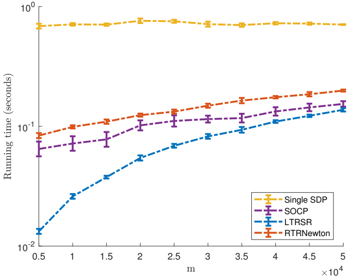


(b) wine data: $\mathcal{A}_{\text{severe}}$

Experiments: Real Datasets



(c) blog dataset: $\mathcal{A}_{\text{modest}}$



(d) blog dataset: $\mathcal{A}_{\text{severe}}$

Experiments: Synthetic Datasets

Table 1: Time (seconds) comparison on dense data

$m = 2n$

m	$\gamma = 0.1$					$\gamma = 0.01$				
	n	SOCP (eig time)	RTRNew	LTRSR	Ratio	n	SOCP (eig time)	RTRNew	LTRSR	Ratio
2000	1000	0.585 (0.064)	0.743	**0.034**	17	1000	0.619 (0.087)	0.712	**0.031**	20
4000	2000	1.957 (0.317)	2.459	**0.177**	11	2000	2.244 (0.419)	2.519	**0.138**	16
8000	4000	10.693 (2.758)	9.269	**0.931**	11	4000	12.123 (3.448)	5.179	**0.956**	13
12000	6000	29.304 (9.444)	18.824	**2.120**	14	6000	33.093 (11.903)	15.857	**2.135**	16
16000	8000	58.561 (21.634)	40.711	**3.982**	15	8000	66.816 (27.466)	38.501	**3.768**	18
20000	10000	114.376 (49.754)	59.768	**6.099**	19	10000	118.044 (49.477)	59.551	**5.916**	20

$m = n$

m	$\gamma = 0.1$					$\gamma = 0.01$				
	n	SOCP (eig time)	RTRNew	LTRSR	Ratio	n	SOCP (eig time)	RTRNew	LTRSR	Ratio
1000	1000	0.454 (0.065)	0.594	**0.017**	27	1000	0.564 (0.086)	0.704	**0.020**	28
2000	2000	2.104 (0.325)	2.600	**0.097**	22	2000	1.900 (0.471)	1.828	**0.098**	19
4000	4000	10.795 (2.698)	6.958	**0.478**	23	4000	11.597 (3.539)	6.789	**0.499**	23
6000	6000	28.391 (9.481)	17.835	**1.083**	26	6000	31.262 (11.691)	18.617	**1.053**	30
8000	8000	55.263 (21.555)	35.510	**2.011**	27	8000	63.983 (27.512)	34.655	**1.984**	32
10000	10000	97.383 (40.091)	58.009	**3.065**	32	10000	109.516 (50.018)	54.060	**3.048**	36

Table 2: Time (seconds) on sparse synthetic data

sparsity =0.01					
m	n	SOCP (eig time)	RTRNew	LTRSR	Ratio
5000	10000	71.601 (39.432)	13.124	**0.225**	318
7500	15000	217.529 (120.456)	26.551	**0.534**	407
10000	20000	513.751 (288.490)	47.411	**1.049**	490
12500	25000	941.394 (539.619)	69.421	**1.606**	586
15000	30000	1539.443 (865.813)	113.223	**2.416**	**637**
sparsity =0.001					
m	n	SOCP (eig time)	RTRNew	LTRSR	Ratio
5000	10000	61.587 (45.253)	1.416	**0.028**	2200
7500	15000	153.075 (117.389)	2.379	**0.053**	2888
10000	20000	335.956 (259.671)	5.453	**0.113**	2973
12500	25000	638.175 (491.391)	7.715	**0.168**	3799
15000	30000	1082.261 (832.413)	12.090	**0.235**	**4605**

Thank You!