Fast Algorithms for Stackelberg Prediction Game with Least Squares Loss

Jiali Wang[1] He Chen[2] Rujun Jiang[1] Xudong Li[1] Zihao Li[2]

Abstract

The Stackelberg prediction game (SPG) has been extensively used to model the interactions between the learner and data provider in the training process of various machine learning algorithms. Particularly, SPGs played prominent roles in cybersecurity applications, such as intrusion detection, banking fraud detection, spam filtering, and malware detection. Often formulated as NP-hard bi-level optimization problems, it is generally computationally intractable to find global solutions to SPGs. As an interesting progress in this area, a special class of SPGs with the least squares loss (SPG-LS) have recently been shown polynomially solvable by a bisection method. However, in each iteration of this method, a semidefinite program (SDP) needs to be solved. The resulted high computational costs prevent its applications for large-scale problems. In contrast, we propose a novel approach that reformulates a SPG-LS as a single SDP of a similar form and the same dimension as those solved in the bisection method. Our SDP reformulation is, evidenced by our numerical experiments, orders of magnitude faster than the existing bisection method. We further show that the obtained SDP can be reduced to a second order cone program (SOCP). This allows us to provide real-time response to large-scale SPG-LS problems. Numerical results on both synthetic and real world datasets indicate that the proposed SOCP method is up to 20,000+ times faster than the state of the art.

1. Introduction

In the big data era, machine learning (ML) algorithms have been extensively used to extract useful information from data and found numerous applications in our daily life. In certain areas, such as cybersecurity, the nature of applications requires high robustness of ML algorithms against adversarial attacks. A typical scenario would be the training data that the ML algorithms or the learner relied on is deliberately altered by a malicious adversary. In this case, the key assumption for the success of ML algorithms, i.e., the stationarity of data or equivalently the independent and identically distributed (i.i.d.) assumption, fails to hold. To alleviate this difficulty, researchers have proposed various game theoretic approaches (Brückner & Scheffer, 2011; Tong et al., 2018; Vorobeychik & Kantarcioglu, 2018; Bishop et al., 2020) to model the strategic interactions between the learner and the attacker – in our case, the adversarial data provider.

In practice, there are also many applications that the learner and the data providers are not entirely antagonistic, where the data providers often manipulate the data only for their own interests. Introduced by Brückner & Scheffer (2011), the SPG is used to model the interactions between the learner and the data provider as a two-players non-zero-sum sequential game for such cases. In the SPG, the learner is regarded as the leader who makes the first move to commit to a predictive model without knowing the strategy of the data provider (or the follower). Then, the data provider, based on the available information of the learner's predictive model, selects his costs-minimizing strategy to modify the data against the learner. Under the rationality assumption of both the learner and data provider, Brückner & Scheffer (2011) introduced the notion of Stackelberg equilibrium as the optimal strategy of the SPG and proposed to find it via solving a corresponding bi-level optimization problem. Particularly, the bi-level optimization problem minimizes the prediction loss from the learner's perspective under the constraint that the data has been optimally modified from the data provider's perspective. Since then, SPGs have received a lot of attention in the literature (Shokri et al., 2012; Zhou & Kantarcioglu, 2016; Wahab et al., 2016; Papernot et al., 2018; Naveiro & Insua, 2019; Zhou et al., 2019). Unfortunately, bi-level optimization problems are generally NP-hard (Jeroslow, 1985) and their optimal solutions are intrinsically difficult to obtain, which severely limit the applicability of SPGs in real world use cases.

Recently, Bishop et al. (2020) made the first step to globally solve a special subclass of SPGs. Specifically, they

[1]School of Data Science, Fudan University, China [2]School of Mathematical Sciences, Fudan University, China. Correspondence to: Rujun Jiang <rjjiang@fudan.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

restricted their interests on SPGs with least squares loss (SPG-LS) (i.e., all the loss functions for the learner and data providers are the least squares). They further reformulated the SPG-LS into a quadratically constrained quadratic fractional program that can be solved via a bisection method. In each iteration of their bisection method, a nonconvex quadratically constrained quadratic program (QCQP) needs to be *exactly* solved. Fortunately, by using the celebrated S-lemma (Yakubovich, 1971; Pólik & Terlaky, 2007; Xia et al., 2016), the optimal solutions to the nonconvex QCQPs can be obtained via solving their semidefinite programming (SDP) relaxations (Vandenberghe & Boyd, 1996). However, the number of bisection searches is often of several tens in practice. This, together with heavy computational costs of solving each SDP, makes the bisection method far less attractive especially for large-scale problems. Moreover, the requirement for *exactly* solving each SDP is too strong for large-scale problems, even armed with powerful academic and commercial solvers. Theoretically speaking, given the accumulation of these inaccuracy, the convergence of the bisection method with inexact SDPs' solutions remains unknown. More importantly, this accumulated inexactness may finally result unstable algorithmic performances, which prevents its applications in the area of security.

In this paper, we aim to resolve the above mentioned scalability and stability issues of the bisection method for the SPG-LS. For this purpose, we start by re-examining the quadratic fractional program (QFP) considered in Bishop et al. (2020). By using the S-lemma in a slightly different way, we show that the QFP can be directly reformulated into an SDP of almost the same problem size as the ones in the bisection method. Furthermore, we prove that there always exists an optimal solution for our SDP and an optimal solution to the SPG-LS can be recovered from the optimal solution of our SDP. It thus implies that the bisection steps are unnecessary, i.e., to solve the SPG-LS, one only needs to solve a single SDP. This novel reformulation outperforms the bisection method by a significant margin as the latter involves solving a series of SDPs with similar problem sizes. Surprisingly, we can take a step further in accelerating our method. By carefully investigating the intrinsic structures of the proposed SDP, we show our single SDP reformulation can be further reduced into a second order cone program (SOCP) (Alizadeh & Goldfarb, 2003), which can be solved much more efficiently than SDPs in general. More specifically, we apply two congruence transformation for the linear matrix inequality (LMI) in our SDP. The second congruence in fact explores a simultaneous diagonalizabiliy of submatrices for the three matrices in the LMI after the first congruence transformation. Then by using a generalized Schur complement, we demonstrate that our SDP can be further reformulated as an SOCP with a much smaller size. The main cost in our reformulation is a spectral decomposition for the data matrix that is cheap even for large instances. Moreover, solving our SOCP reformulation is even cheaper than one spectral decomposition. Hence our SOCP method is much faster than our single SDP method.

We summarise our contributions as follows:

- We derive a single SDP reformulation for the SPG-LS, while the state of the art needs to solve dozens of SDPs with similar problems sizes.
- We further derive an SOCP reformulation with a much smaller dimension than our single SDP.
- We propose two efficient ways to recover an optimal solution for the SPG-LS either from a rank-1 decomposition of the dual solution of our single SDP or by solving a linear system with an additional equation.
- We show that our methods significantly improve the state of the art by numerical experiments on both synthetic and real data sets.

2. Preliminaries

In this section, we formalize the SPG-LS problem by adapting the same setting as in Bishop et al. (2020). A brief review of Bishop et al.'s bisection method will also be provided.

Similar as in Bishop et al. (2020), we assume that the learner has access to a sample $S = \{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^m$ with each $\mathbf{x}_i \in \mathbb{R}^n$ been the input example, y_i and z_i been the output labels of interests to the learner and the data provider, respectively. The samples are assumed to be realizations of $(\mathbf{x}, y, z)$ following some fixed but unknown distribution $\mathcal{D}$. The learner then aims to train a linear predictor $\mathbf{w} \in \mathbb{R}^n$ based on S, i.e., to predict correctly label y when supplied with $\mathbf{x}$. In the SPG-LS, being aware of the learner's predictor $\mathbf{w}$, the goal of the adversarial data provider is to fool the learner to predict the label z by modifying the input data $\mathbf{x}$ to $\hat{\mathbf{x}}$ while maintaining low manipulation costs. Here, we follow Bishop et al. (2020) to model the modifying costs from $\mathbf{x}$ to $\hat{\mathbf{x}}$ as $\gamma\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ with some positive parameter γ.

To find the Stackelberg equilibrium of the above SPG-LS, we formulate in the following the corresponding bi-level optimization problem. Given the disclosed predictor $\mathbf{w} \in \mathbb{R}^n$ and the training set S, the data provider described above aims to solve the following optimization problems:

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

Then, one can obtain a Stackelberg equilibrium through the classic backward induction procedure (Brückner & Scheffer, 2011). With the modified data $\{\mathbf{x}_i^*\}_{i=1}^m$, the learner has to solve the following optimization problem:

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

The predictor $\mathbf{w}^*$ and the optimal modified data sets $\{\mathbf{x}_i^*\}_{i=1}^m$ of the data provider are by definition a Stackelberg equilibrium (Brückner & Scheffer, 2011). To obtain this, we arrive at the following bi-level optimization problem:

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

where the i-th row of $X \in \mathbb{R}^{m\times n}$ is just the input example $\mathbf{x}_i$ and the i-th entries of $\mathbf{y}, \mathbf{z} \in \mathbb{R}^m$ are labels y_i and z_i, respectively.

In their work, Bishop et al. (2020) considered the following reformulation. They started by replacing the lower differentiable and strongly convex optimization problem by its optimality condition, i.e.,

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

Then, the Sherman-Morrison formula (Sherman & Morrison, 1950) further implies

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

Substituting the above formula to problem (1), we obtain the following fractional program:

$$\min_{\mathbf{w}} \frac{\left\|\frac{1}{\gamma}\mathbf{z}\mathbf{w}^T\mathbf{w} + X\mathbf{w} - \mathbf{y} - \frac{1}{\gamma}\mathbf{w}^T\mathbf{w}\mathbf{y}\right\|^2}{(1 + \frac{1}{\gamma}\mathbf{w}^T\mathbf{w})^2}. \tag{2}$$

2.1. A Bisection Method for Solving (2)

Here, we briefly review the bisection method developed in Bishop et al. (2020) for solving the fractional program (2).

By introducing an artificial variable α and an additional constraint $\alpha = \mathbf{w}^T\mathbf{w}$, Bishop et al. (2020) first reformulated (2) as the following QFP:

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

Then, they adopted a bisection search for q^* such that $F(q^*) = 0$, where F is the optimal value function of the following Dinkelbach problem associated with (3), for all $q \in \mathbb{R}$,

$$\begin{aligned} F(q) := \min_{\mathbf{w},\alpha} \quad & \|\frac{\alpha}{\gamma}\mathbf{z} + X\mathbf{w} - \mathbf{y} - \frac{\alpha}{\gamma}\mathbf{y}\|^2 - q(1 + \frac{\alpha}{\gamma})^2 \\ \text{s.t} \quad & \alpha = \mathbf{w}^T\mathbf{w}. \end{aligned} \tag{4}$$

The correctness of their algorithm is due to the following well known result for fractional programming.

Lemma 2.1 (Theorem 1 of Dinkelbach (1967)) *Assume that for all $q \in \mathbb{R}$, problem (4) has nonempty optimal solution set. Then, the equation $F(q) = 0$ has a unique solution. Furthermore, $(\mathbf{w}^*, \alpha^*)$ is a solution to the QFP (3) if and only if $\mathbf{w}^{*T}\mathbf{w}^* = \alpha^*$ and $F(q^*) = 0$ where $q^* = \|\frac{\alpha^*}{\gamma}\mathbf{z} + X\mathbf{w}^* - \mathbf{y} - \frac{\alpha^*}{\gamma}\mathbf{y}\|^2/(1 + \alpha^*/\gamma)^2$.*

As $F(q)$ is a concave monotonically decreasing continuous function (Dinkelbach, 1967), the bisection algorithm is well-defined. Bishop et al. (2020) further showed that initial lower and upper bounds q_1 and q_2 for q^* satisfying $F(q_1) \geq 0$ and $F(q_2) \leq 0$ are also easy to obtain.

In each iteration of the bisection method, given q, one needs to compute $F(q)$, i.e., the nonconvex optimization problem (4) needs to be solved. To this purpose, Bishop et al. (2020) applied the S-lemma with equality (Xia et al., 2016) to transform the QCQP (4) into an SDP problem whose optimal objective is exactly $F(q)$. More specifically, define matrices

$$\hat{A} = \begin{pmatrix} X^TX & \frac{1}{\gamma}X^T(\mathbf{z}-\mathbf{y}) & -X^T\mathbf{y} \\ \frac{1}{\gamma}(\mathbf{z}-\mathbf{y})^TX & \frac{1}{\gamma^2}\|\mathbf{z}-\mathbf{y}\|^2 & -\frac{1}{\gamma}(\mathbf{z}-\mathbf{y})^T\mathbf{y} \\ -\mathbf{y}^TX & -\frac{1}{\gamma}\mathbf{y}^T(\mathbf{z}-\mathbf{y}) & \mathbf{y}^T\mathbf{y} \end{pmatrix},$$

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

Given $q \in \mathbb{R}$, problem (4) admits the same objective value with the following SDP

$$\max_{\tau,\lambda} \ \tau \quad \text{s.t. } \hat{A} - q\hat{B} + \lambda\hat{C} - \tau E \succeq 0, \tag{5}$$

where $E = \text{Diag}\,(0_{n+1}, 1) \in \mathbb{R}^{(n+2)\times(n+2)}$ is the diagonal matrix with first $n+1$ diagonal entries being zero and the last entry being one. Then, the SDP (5) is solved by advanced interior point methods (IPM).

Theoretically, Bishop et al. (2020) showed that under the assumption that each involved SDP is solved exactly, the bisection method needs $\log_2(2\mathbf{y}^T\mathbf{y}/\varepsilon)$ steps to obtain an ε-optimal estimation of q^* with given tolerance $\varepsilon > 0$. Note that in practice, $\mathbf{y}^T\mathbf{y}$ can be quite large and ε may be required to be small. Thus, the bisection method may need to solve a significant numbers of SDPs even in the moderate-scale setting, e.g., the numbers of samples m and features n are several thousands. Since the amount of work per iteration of IPM for solving (5) is $\mathcal{O}(n^3)$ (Nesterov & Nemirovskii, 1993; Todd, 2001), the total computational costs of the bisection method can be prohibitive. Moreover, there in fact exists no optimization solver which can return exact solutions to these SDPs. Hence, the convergence theory of the bisection method may break down and its stability may be implicitly affected due to the accumulation of optimization errors in each iteration. These issues on scalability and stability of the bisection method motivate our study in this paper.

3. Single SDP Reformulation

In this section, we present a novel result that shows an optimal solution to (2), or the Stackelberg equilibrium of the SPG-LS, can be obtained by just solving a single SDP with a similar size as the SDP (5). To begin, let us consider the following equivalent formulation of (2):

$$\min_{\mathbf{w},\,\alpha} \quad \frac{\|\alpha\mathbf{z} + X\mathbf{w} - \mathbf{y} - \alpha\mathbf{y}\|^2}{(1+\alpha)^2} \qquad \text{s.t.} \quad \frac{\mathbf{w}^T\mathbf{w}}{\gamma} = \alpha, \tag{6}$$

which is slightly different from (3) in a scaling of α. Now let us recall the following *S-lemma with equality*, which is the main tool in showing the equivalence of (4) and (5) in Bishop et al. (2020).

Lemma 3.1 (Theorem 3 in Xia et al. (2016)) *Let $f(\mathbf{x}) = \mathbf{x}^T Q_1 \mathbf{x} + 2\mathbf{p}_1^T\mathbf{x} + q_1$ and $h(\mathbf{x}) = \mathbf{x}^T Q_2 \mathbf{x} + 2\mathbf{p}_2^T\mathbf{x} + q_2$, where $Q_1, Q_2 \in \mathbb{R}^{n\times n}$ are symmetric matrices, $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^n$ and $q_1, q_2 \in \mathbb{R}$. If function h takes both positive and negative values and $Q_2 \neq 0$, then following two statements are equivalent:*

1. *There is no $\mathbf{x} \in \mathbb{R}^n$ such that $f(\mathbf{x}) < 0$, $h(\mathbf{x}) = 0$.*
2. *There exists a $\lambda \in \mathbb{R}$ such that $f(\mathbf{x}) + \lambda h(\mathbf{x}) \geq 0$.*

We also need the following result that is well known in quadratic programming.

Lemma 3.2 (Theorem 2.43 in Beck (2014)) *Let $Q \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $\mathbf{p} \in \mathbb{R}^n$ and $q \in \mathbb{R}$. Then the following two statements are equivalent:*

1. $(\mathbf{x}^T, 1)\begin{pmatrix} Q & \mathbf{p} \\ \mathbf{p}^T & q \end{pmatrix}\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \geq 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$.
2. $\begin{pmatrix} Q & \mathbf{p} \\ \mathbf{p}^T & q \end{pmatrix} \succeq 0$.

From now on, let us define

$$\begin{aligned} A &= \begin{pmatrix} X^TX & X^T(\mathbf{z}-\mathbf{y}) & -X^T\mathbf{y} \\ (\mathbf{z}-\mathbf{y})^TX & \|\mathbf{z}-\mathbf{y}\|^2 & -(\mathbf{z}-\mathbf{y})^T\mathbf{y} \\ -\mathbf{y}^TX & -\mathbf{y}^T(\mathbf{z}-\mathbf{y}) & \mathbf{y}^T\mathbf{y} \end{pmatrix}, \\ B &= \begin{pmatrix} \mathbf{0}_n & & \\ & 1 & 1 \\ & 1 & 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} \frac{I_n}{\gamma} & & \\ & 0 & -\frac{1}{2} \\ & -\frac{1}{2} & 0 \end{pmatrix}. \end{aligned} \tag{7}$$

With the above facts, we are now ready to present our main result of this section that (2) can be equivalently reformulated as a single SDP, where our SDP reformulation follows a similar idea in equations (1.12-1.14) in (Nguyen et al., 2014).

Theorem 3.3 *Problem (2) is equivalent to the following SDP*

$$\begin{array}{ll} \sup_{\mu,\lambda} & \mu \\ \text{s.t.} & A - \mu B + \lambda C \succeq 0. \end{array} \tag{8}$$

Proof. Consider the equivalent formulation (6). Let $f(\mathbf{w}, \alpha) = \|\alpha\mathbf{z} + X\mathbf{w} - \mathbf{y} - \alpha\mathbf{y}\|^2$, $p(\mathbf{w}, \alpha) = \frac{\mathbf{w}^T\mathbf{w}}{\gamma} - \alpha$ and denote by v_{frac} the optimal value of (6). Recall the definitions of A, B, and C in (7). Then, we conduct the reformulation in the following manner:

$$\begin{aligned} v_{\text{frac}} &= \inf_{\mathbf{w},\alpha} \left\{ \frac{f(\mathbf{w},\alpha)}{(1+\alpha)^2} : p(\mathbf{w},\alpha) = 0 \right\} \\ &= \sup_{\mu} \left\{ \mu : \begin{array}{l} \{(\mathbf{w},\alpha) \mid f(\mathbf{w},\alpha) - \mu(1+\alpha)^2 < 0, \\ p(\mathbf{w},\alpha) = 0\} = \emptyset \end{array} \right\} \\ &= \sup_{\mu} \left\{ \mu : \begin{array}{l} \exists\lambda \in \mathbb{R} \text{ s.t.} f(\mathbf{w},\alpha) - \mu(1+\alpha)^2 \\ +\lambda p(\mathbf{w},\alpha) \geq 0,\ \forall \mathbf{w} \in \mathbb{R}^n, \alpha \in \mathbb{R} \end{array} \right\} \end{aligned} \tag{9}$$

$$\begin{aligned} &= \sup_{\mu,\lambda} \left\{ \mu : \begin{array}{l} \left(\mathbf{w}^T \ \alpha \ 1 \right)(A - \mu B + \lambda C)\begin{pmatrix} \mathbf{w} \\ \alpha \\ 1 \end{pmatrix} \geq 0, \\ \forall \mathbf{w} \in \mathbb{R}^n, \alpha \in \mathbb{R} \end{array} \right\} \\ &= \sup_{\mu,\lambda} \left\{ \mu : A - \mu B + \lambda C \succeq 0 \right\}, \end{aligned} \tag{10}$$

where (9) is due to the S-lemma with equality in Lemma 3.1 and (10) is due to Lemma 3.2. □

We briefly remark that there exists an optimal solution for the SDP (8) and it can be used to recover an optimal solution to (2). In fact, we can recover an optimal solution to (2) by either doing a rank-1 decomposition for the dual solution of SDP (8), thanks to Sturm & Zhang (2003), or solving a linear system with an additional equation as in step 8 in Algorithm 1[1]. More details are given in Appendix.

Up to now, we have shown that to obtain a global optimal solution to the nonconvex fractional program (2), only a single SDP needs to be solved. A crucial observation is that our single SDP (8) has a similar form and the same dimension of the matrices with (5), the subproblem in each iteration of the bisection method. We remark the main differences: (i) the bisection parameter q is the variable μ in our formulation; (ii) our formulation does not involve a τ which is used for generating new half interval in the bisection method. From the similar forms of two SDPs, we can expect that solving the SDPs (8) and (5) needs a similar CPU time. However, the bisection method needs to solve a series of SDPs. Indeed, for each test instance in our numerical experiments, the bisection method needs to solve about 30 SDPs. In other words, our single SDP method is a more efficient way to obtain q^* (or equivalently, μ in (8)) such that $F(q^*) = 0$, which closely relates to an optimal

[1] See the discussions after Theorem 4.1.

solution of problem (3) (or equivalently, problem (2)) in view of Lemma 2.1[2], than the bisection method.

4. SOCP Reformulation

Though our single SDP approach introduced in the previous section for finding Stackelberg equilibrium of SPG-LS has already been much faster than the bisection method, the fact that solving a large-scale SDP requires extensive computations motivates us to make a step further of seeking more reductions. For this purpose, in this section, by using a simultaneous diagonalizability of submatrices in the linear matrix inequality (LMI) constraint in (8), we can further reformulate SDP (8) as an SOCP that can be solved much more efficiently. We briefly describe our main idea in Algorithm 1.

Algorithm 1 SOCP method for solving (2)

```
Input: matrices $A, B, C$ in (7)
set $V_1$ as in (11)
set $\bar{A}, \bar{B}, \bar{C}$ as in (14), (12), (13)
do spectral decomposition to matrix $\bar{A}_{11}$ in (14) with $\bar{A}_{11} = HDH^T$
set $V_2$ as in (15)
obtain the matrices $\tilde{A} = V_2\bar{A}V_2, \tilde{B} = V_2\bar{B}V_2, \tilde{C} = V_2\bar{C}V_2$ in forms (16) and (17) with diagonal $n+1$th order leading principal submatrices
solve the SOCP problem (20) to obtain optimal $\mu^*, \lambda^*$
obtain $\mathbf{w}^*$ by finding a solution of the following linear system
    $$(A - \mu^* B + \lambda^* C) \begin{pmatrix} \mathbf{w} \\ \alpha \\ 1 \end{pmatrix} = 0$$
    satisfying $\frac{1}{\gamma}\mathbf{w}^T\mathbf{w} = \alpha$
```

The motivation of our reformulation comes from simple observations on matrices A, B and C. The first key observation is that B and C can be simultaneously diagonalized by congruence. Indeed, letting

$$V_1 = \begin{pmatrix} I_n & 0 & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} & 1 \\ 0 & -\frac{1}{\sqrt{\gamma}} & 1 \end{pmatrix}, \tag{11}$$

we have from (7)

$$\bar{B} := V_1^T B V_1 = \begin{pmatrix} \mathbf{0}_{n+1} & \\ & 4 \end{pmatrix}, \tag{12}$$

and

$$\bar{C} := V_1^T C V_1 = \begin{pmatrix} \frac{1}{\gamma} I_{n+1} & \\ & -1 \end{pmatrix}. \tag{13}$$

For convenience, let

$$\bar{A} := V_1^T A V_1 = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{12}^T & \bar{A}_{22} \end{pmatrix}. \tag{14}$$

The second key observation is that the $n+1$th order leading principal submatrices of A, B and C can be simultaneously diagonlizable by congruence. To see this, applying spectral decomposition to the real symmetric matrix $\bar{A}_{11}$ yields $\bar{A}_{11} = HDH^T$, where H is an $(n+1) \times (n+1)$ orthogonal matrix and $D = \mathrm{Diag}\,(\mathbf{d})$ is a diagonal matrix with d_i being its ith diagonal entry. Define

$$V_2 = \begin{pmatrix} H & 0 \\ 0 & 1 \end{pmatrix}. \tag{15}$$

Now we have

$$\tilde{A} := V_2^T \bar{A} V_2 = \begin{pmatrix} D & \mathbf{b} \\ \mathbf{b}^T & c \end{pmatrix}, \tag{16}$$

where $\mathbf{b} \in \mathbb{R}^{n+1}$ and $c \in \mathbb{R}$. Since $H^T H = I$, we also have

$$\tilde{B} := V_2^T \bar{B} V_2 = \bar{B} \quad \text{and} \quad \tilde{C} = V_2^T \bar{C} V_2 = \bar{C}. \tag{17}$$

As V_1 and V_2 are both invertible matrices, the LMI constraint in (8) is equivalent to

$$\tilde{A} - \mu\tilde{B} + \lambda\tilde{C} \succeq 0. \tag{18}$$

From the generalized Schur complement (Zhang, 2006), the LMI (18) is equivalent to

$$\begin{aligned} & D + \tfrac{\lambda}{\gamma} I_{n+1} \succeq 0, \\ & \mathbf{b} \in \mathrm{Range}(D + \tfrac{\lambda}{\gamma} I_{n+1}), \\ & c - 4\mu - \lambda - \mathbf{b}^T (D + \tfrac{\lambda}{\gamma} I_{n+1})^\dagger \mathbf{b} \succeq 0, \end{aligned} \tag{19}$$

where $(M)^\dagger$ denotes the Moore-Penrose pseudoinverse of matrix M. As D is a diagonal matrix, by defining $\frac{0}{0} = 0$, (19) is further equivalent to

$$\begin{aligned} & d_i + \tfrac{\lambda}{\gamma} \geq 0, \text{ and } b_i = 0 \text{ if } d_i + \tfrac{\lambda}{\gamma} = 0,\ i \in [n+1], \\ & c - 4\mu - \lambda - \textstyle\sum_{i=1}^{n+1} \frac{b_i^2}{d_i + \lambda/\gamma} \geq 0. \end{aligned}$$

These constraints can be further rewritten as

$$\begin{aligned} & d_i + \tfrac{\lambda}{\gamma} \geq 0,\ i \in [n+1], \\ & c - 4\mu - \lambda - \textstyle\sum_{i=1}^{n+1} s_i \geq 0, \\ & s_i(d_i + \tfrac{\lambda}{\gamma}) \geq b_i^2,\ i \in [n+1]. \end{aligned}$$

For $i \in [n+1]$, each constraint $s_i(d_i + \frac{\lambda}{\gamma}) \geq b_i^2$ can be expressed as a rotated second order cone constraint (Alizadeh & Goldfarb, 2003), which is equivalent to the second order cone constraint

$$\sqrt{b_i^2 + \left(\frac{s_i + d_i - \frac{\lambda}{\gamma}}{2}\right)^2} \leq \frac{s_i + d_i + \frac{\lambda}{\gamma}}{2}.$$

Consequently, we have the following theorem.

[2] In fact, we use Lemma 2.1 slightly different from its original statement with a scaling of γ here and in the discussions after Theorem 4.1.

Theorem 4.1 *With the same notation in this section, problem (8) is equivalent to the following SOCP problem*

$$\begin{array}{ll} \sup_{\mu,\lambda,\mathbf{s}} & \mu \\ \text{s.t.} & d_i + \frac{\lambda}{\gamma} \geq 0,\ i \in [n+1], \\ & c - 4\mu - \lambda - \sum_{i=1}^{n+1} s_i \geq 0, \\ & s_i(d_i + \frac{\lambda}{\gamma}) \geq\ b_i^2,\ i \in [n+1]. \end{array} \tag{20}$$

Note that based on our construction, i.e., the congruence transformations to the matrices in the LMI constraint in (8), any optimal solution (μ^*, λ^*) to (20) is still optimal to (8). We claim that an optimal solution $\mathbf{w}^*$ to (2) can be recovered by solving the linear system with an additional equation in step 8 in Algorithm 1. Indeed, Lemma 2.1, the strong duality theory of SDPs and the S-lemma with equality guarantee the existence of a rank-1 solution to the SDP relaxation of

$$\begin{array}{ll} \min\limits_{\mathbf{w},\alpha} & \|\alpha\mathbf{z} + X\mathbf{w} - \mathbf{y} - \alpha\mathbf{y}\|^2 - \mu^*(1 + \dfrac{\alpha}{\gamma})^2 \\ \text{s.t} & \dfrac{1}{\gamma}\mathbf{w}^T\mathbf{w} = \alpha, \end{array}$$

and the solution solves the following KKT system of the corresponding SDP relaxation

$$\begin{cases} \langle C, W\rangle = 0, \\ W_{n+2,n+2} = 1, \\ W \succeq 0, \\ \langle A - \mu^* B + \lambda^* C, W\rangle = 0. \end{cases}$$

By setting $W = \begin{pmatrix} \mathbf{w} \\ \alpha \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{w}^T & \alpha & 1 \end{pmatrix}$, the above facts are equivalent to

$$(A - \mu^* B + \lambda^* C)\begin{pmatrix} \mathbf{w} \\ \alpha \\ 1 \end{pmatrix} = 0, \quad \frac{1}{\gamma}\mathbf{w}^T\mathbf{w} = \alpha,$$

due to $A - \mu^* B + \lambda^* C \succeq 0$. One may think that the above equations are difficult to solve. In fact, the linear system usually only has a unique solution and it suffices to solve the linear system solely. A sufficient condition to guarantee this is that the matrix $(A - \mu^* B + \lambda^* C)$ is of rank $n+1$, which is exactly the case in all our numerical tests. More discussions on the solution recovering are given in Appendix.

In general, SOCPs can be solved much faster than SDPs. For our problem, it can be seen that IPMs for solving SOCP (20) takes $\mathcal{O}(n)$ costs per iteration (Alizadeh & Goldfarb, 2003; Andersen et al., 2003; Tütüncü et al., 2003) which is of orders magnitudes faster than the case $\mathcal{O}(n^3)$ in solving SDP (8) using interior point methods. The high efficiency of our SOCP approach is also evidenced by our numerical tests.

5. Experiment Results

In this section, we conduct numerical experiments on both synthetic and real world datasets to verify the superior performance of our proposed algorithms in terms of both the computational time and the learning accuracy. We apply the powerful commercial solver MOSEK (MOSEK, 2021) to solve all the SDPs and SOCPs in the bisection method and ours.

All simulations are implemented using MATLAB R2019a on a PC running Windows 10 Intel(R) Xeon(R) E5-2650 v4 CPU (2.2GHz) and 64GB RAM. We report the results of two real datasets and three synthetic datasets and defer other results to the supplementary material.[3]

5.1. Real World Dataset

We first demonstrate the accuracy and efficiency of our proposed methods on two real datasets. We compare the average mean squared error (MSE) as well as the wall-clock time of our SDP and SOCP approaches with those of the bisection method in Bishop et al. (2020), the ridge regression and a nonlinear programming reformulation of the SPG-LS in Brückner & Scheffer (2011). Similar as in Bishop et al. (2020), to evaluate the learning accuracy of the algorithms, we perform 10-fold cross-validation and compare their average MSE for 40 different values of the parameter $\gamma \in [1 \times 10^{-3}, 0.75]$ in (2). For each γ, a grid search on 9 logarithmically spaced points $[1 \times 10^{-5}, 1000]$ is used to compute the best regularization parameter for the ridge regression. We also compare the running time of all the methods at $\gamma = 0.5$, averaged over 10 trials to further illustrate the efficiency of our methods. For the testing purpose, we first apply min-max normalization to the raw data X and scale the labels y, z to $y = y/(\beta\|y\|_\infty)$ and $z = z/(\beta\|y\|_\infty)$, respectively. These labels will be scaled back to compute the average MSE. It is worth noting that the constant β can be adjusted with respect to different datasets.

5.1.1. Wine Dataset

We first test our methods on the red wine dataset (Cortez et al., 2009), which contains 1599 instances each with 11 features. The response is a physiochemical measurement ranged from 0 to 10, where higher score means better quality. We use the same setting as in Bishop et al. (2020). The wine provider manipulates the data to achieve a higher score if the original label is smaller than some threshold t. The wine provider sets his target label z as follows,

$$z_i = \max\{y_i, t\}.$$

We consider two different providers $\mathcal{A}_{\text{modest}}$ with $t_{\text{modest}} = 6$ and $\mathcal{A}_{\text{severe}}$ with $t_{\text{severe}} = 8$.

Our numerical results are reported in Figure 1. From Figures 1(a) and 1(b), we see that our single SDP method, our SOCP method and the bisection method achieved the best

[3] Our code is available at `https://github.com/JialiWang12/SPGLS`.

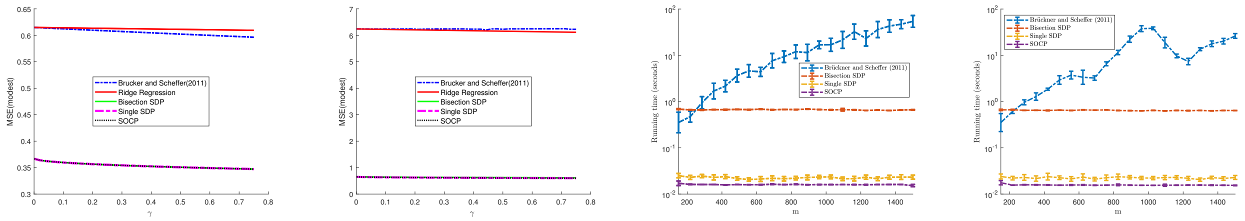

Figure 1. Performance comparison between different algorithms on the red wine dataset. The left two plots correspond to MSE result generated by $\mathcal{A}_{\text{modest}}$ and $\mathcal{A}_{\text{severe}}$, whilst the right two plots correspond to wall-clock time comparison generated by $\mathcal{A}_{\text{modest}}$ and $\mathcal{A}_{\text{severe}}$.

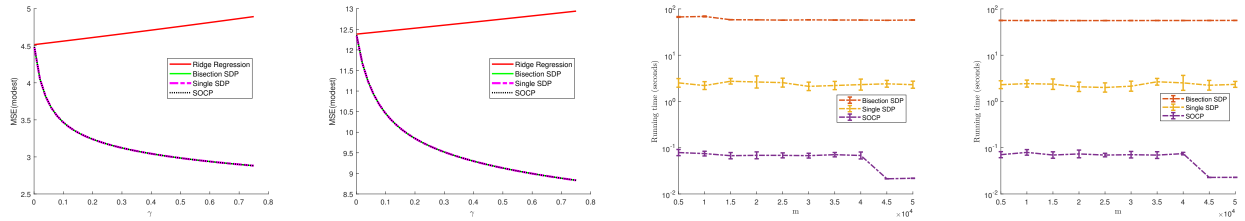

Figure 2. Performance comparison between different algorithms on the blog dataset. The left two plots correspond to MSE result generated by $\mathcal{A}_{\text{modest}}$ and $\mathcal{A}_{\text{severe}}$, whilst the right two plots correspond to wall-clock time comparison generated by $\mathcal{A}_{\text{modest}}$ and $\mathcal{A}_{\text{severe}}$.

performance in average MSE. This is no surprise as the three methods are guaranteed to solve the SPG-LS globally. Figures 1(c) and 1(d) indicate that both single SDP and SOCP are much faster than all the other three methods. Since the dimension of SDP is rather small, our SDP method took a similar time with our SOCP method.

5.1.2. Blog Dataset

We next compare our algorithms on the blogfeedback dataset[4] from the UCI data repository (Dua & Graff, 2019). It consists of 52397 data processed from raw feedback-materials collected from the Internet. Each one conveys the information of a certain session, described by 281 features. The response is the number of comments. The task for the learner, in this case, is to predict the the future comment numbers in a regression manner.

As before, we assume that the label $z_i = \max\{y_i + \delta, 0\}$ is modified by the data provider in order to trigger a biased result. For example, consider an option guider who aims to manipulate the public expectation of a certain blog news. He is then motivated to temper the announced comment number. We assume there are two types of data providers, $\mathcal{A}_{\text{modest}}$ with $\delta = -5$ and $\mathcal{A}_{\text{severe}}$ with $\delta = -10$. All the other hyperparameters are the same with the wine dataset.

For this dataset, we do not compare the method in Brückner & Scheffer (2011) for time consideration. Hence we only present the comparisons of the other four methods in Figure 2. Similarly, Figures 2(a) and 2(b) demonstrate that our two methods achieved the best average MSE. Figures 2(c) and 2(d) indicate that both the single SDP and SOCP methods are much faster than the bisection method, and the SOCP approach surpasses the single SDP approach. In fact, our SOCP method takes only about $1/50$ time of our single SDP method, while our single SDP method takes only about $1/20$ time of the bisection method. That is, our SOCP method is 1000 times more efficient than the bisection method on this dataset.

5.2. Synthetic Dataset

To further demonstrate the efficacy of our proposed approaches in large-scale problems, we perform synthetic experiments with a high feature dimension. The function `make_regression` in scikit-learn (Pedregosa et al., 2011) is used to build artificial datasets of controlled size and complexity. In particular, we specify the noise as 0.1, which is the standard deviation of the Gaussian noise applied to the output y, and all other arguments are set as default. Our experiment focuses on the comparison of three different methods including the bisection method, the single SDP and SOCP methods. Similar as in Bishop et al. (2020), the fake input label z_i is set as

$$z_i = \max\{y_i, y_{0.25}\},$$

where $y_{0.25}$ represents the lower quartile (25th percentile) of output y. More specifically, if the true label is greater

[4] https://archive.ics.uci.edu/ml/datasets/BlogFeedback

than or equal to the threshold $y_{0.25}$, then the label would not be modified. Otherwise, the label would be set as $y_{0.25}$. In all tests, the parameter γ is set as 0.01. More results with $\gamma = 0.1$ can be found in the Appendix.

Table 1. Time (seconds) comparison on synthetic data: $m = 2n$

m	n	bisect	sSDP	SOCP	ratio1	ratio2	eig
200	100	4.356	0.111	0.043	101	3	0.001
1000	500	167.732	3.997	0.099	1702	41	0.020
2000	1000	988.675	45.984	0.178	5559	259	0.085
4000	2000	7877.041	438.487	0.536	14694	818	0.441
8000	4000	-	3127.316	1.478	-	2116	3.349
12000	6000	-	-	3.079	-	-	11.245

Table 2. Time (seconds) comparison on synthetic data: $m = n$

m	n	bisect	sSDP	SOCP	ratio1	ratio2	eig
100	100	4.342	0.107	0.040	108	3	0.001
500	500	158.304	4.142	0.072	2197	57	0.018
1000	1000	990.151	21.781	0.225	4408	97	0.085
2000	2000	7667.927	201.411	0.586	13094	344	0.442
4000	4000	-	2142.952	2.485	-	862	3.264
6000	6000	-	-	2.876	-	-	11.117

Table 3. Time (seconds) comparison on synthetic data: $m = 0.5n$

m	n	bisect	sSDP	SOCP	ratio1	ratio2	eig
50	100	4.146	0.105	0.047	87	2	0.001
250	500	156.018	4.471	0.078	2004	57	0.021
500	1000	956.343	69.267	0.189	5047	366	0.080
1000	2000	7495.735	177.999	0.371	20217	480	0.405
2000	4000	-	1485.843	1.229	-	1209	3.144
3000	6000	-	8769.430	2.616	-	3352	10.436

Tables 1, 2 and 3 summarise the comparison of wall-clock time on different scales with $m = pn,\ p \in \{0.5, 1, 2\}$. In these tables, "bisect" represents the bisection method in Bishop et al. (2020), "sSDP" represents our single SDP method, "SOCP" represents our SOCP method, "ratio1" represents the ratio of times of the bisection method and our SOCP method, and "ratio2" represents the ratio of times of our single SDP method and our SOCP method. The last column "eig" recorded the spectral decomposition time of matrix $\bar{A}_{11}$ in (14). In the test, the algorithm would not be run in larger dimension case (denoted by "-"), if its wall-clock time at current dimension exceeds 1800 seconds.

From the three tables, we can find that our single SDP method is consistently faster than the bisection method. The ratios in the table also demonstrate the high efficiency of our SOCP method, which can be up to 20,000+ times faster than the bisection method for case $(m, n) = (1000, 2000)$. Our SOCP method is also significantly faster than our single SDP method. For example, our SOCP method took about 3 seconds for all cases with $n = 6000$, while our single SDP method took at least 8,000 seconds for the case $(m, n) = (3000, 6000)$. We also remark that the performance gap grows considerably with the problem size since both the ratios increase as the dimension increases. Finally, we mention that, compared to the time of our single SDP method, the time of spectral decomposition in formulating our SOCP is rather small, which is about 11 seconds for $n = 6000$.

6. Conclusion

In this paper, we study the computation for Stackelberg equilibrium of SPG-LSs. Hidden convexity in the fractional programming formulation (2) of the SPG-LS is deeply explored. Then, we are able to reformulate the SPG-LS as a single SDP, based on the S-lemma with equality. By using simultaneous diagonalizability of its submatrices in the constraint, we further reformulate our SDP into an SOCP. We also demonstrate the optimal solution to the SPG-LS can be recovered easily from solving our obtained SDP or SOCP. Numerical comparisons between our single SDP and SOCP approaches with the state of the art demonstrate the high efficiency as well as learning accuracy of our methods for handling large-scale SPG-LSs. We believe that our work opens up a new way for the applicability of SPG-LSs in large-scale real scenarios.

Acknowledgements

Rujun Jiang is partly supported by Natural Science Foundation of China (NSFC) 11801087. Xudong Li is partly supported by the Young Elite Scientists Sponsorship Program by CAST (2019QNRC001).

References

Alizadeh, F. and Goldfarb, D. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003.

Andersen, E. D., Terlaky, T., and Roos, C. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(3):249-277, 2003.

Beck, A. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 2014.

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

Brückner, M. and Scheffer, T. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 547–555, 2011.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547 – 553, 2009.

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

Dua, D. and Graff, C. UCI machine learning repository http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science, 2019.

Jeroslow, R. G. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32(2):146–164, 1985.

MOSEK Aps. The MOSEK optimization toolbox for MATLAB manual. Version 9.2.36. http://docs.mosek.com/9.0/toolbox/index.html, 2021.

Naveiro, R. and Insua, D. R. Gradient methods for solving stackelberg games. In *International Conference on Algorithmic Decision Theory*, pp. 126–140. Springer, 2019.

Nesterov, Y. and Nemirovskii, A. *Interior-point polynomial algorithms in convex programming*. SIAM Studies in Applied Mathematics, Philadelphia, 1993.

Nguyen, V.-B., Sheu, R.-L., and Xia, Y. An sdp approach for solving quadratic fractional programming problems, 2014.

Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. P. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 399–414, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

Pólik, I. and Terlaky, T. A survey of the S-lemma. *SIAM Review*, 49(3):371–418, 2007.

Sherman, J. and Morrison, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21 (1):124–127, 1950.

Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.-P., and Le Boudec, J.-Y. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and Communications Security*, pp. 617–627, 2012.

Sturm, J. F. and Zhang, S. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

Todd, M. J. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

Tong, L., Yu, S., Alfeld, S., et al. Adversarial regression with multiple learners. In *International Conference on Machine Learning*, pp. 4946–4954. PMLR, 2018.

Tütüncü, R. H., Toh, K.-C., and Todd, M. J., Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2): 189–217, 2003.

Vandenberghe, L. and Boyd, S. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

Vorobeychik, Y. and Kantarcioglu, M. Adversarial machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–169, 2018.

Wahab, O. A., Bentahar, J., Otrok, H., and Mourad, A. A stackelberg game for distributed formation of business-driven services communities. *Expert Systems with Applications*, 45:359–372, 2016.

Xia, Y., Wang, S., and Sheu, R.-L. S-lemma with equality and its applications. *Mathematical Programming*, 156 (1-2):513–547, 2016.

Yakubovich, V. A. S-Procedure in Nonlinear Control Theory. *Vestnik Leningrad University*, 1:62–77, 1971.

Zhang, F. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Zhou, Y. and Kantarcioglu, M. Modeling adversarial learning as nested stackelberg games. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 350–362. Springer, 2016.

Zhou, Y., Kantarcioglu, M., and Xi, B. A survey of game theoretic approach for adversarial machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1259, 2019.