A Game Theoretic Approach to Robust Optimization

Sayak Ray Chowdhury
ME-SSA
Department of Electrical Engineering
Indian Institute of Science
sayakrc@ee.iisc.ernet.in

18.06.2015

Abstract

Robust Optimization is a common framework in optimization under uncertainty when the problem parameters are not known, but it is rather known that the parameters belong to some given uncertainty set. In this framework the problem solved is a min-max problem, where the solution is obtained considering the worst possible realization of parameters. However, the problem eventually solved becomes more complicated and sometimes gets computationally intractable as the dimension of the problem increases. For example, solving a robust conic quadratic program with ellipsoidal uncertainty leads to a semidefinite program. Here we will see a general framework for approximately solving a robust optimization problem using tools from online convex optimization and game theory. We formulate the robust optimization problem as a two player zero-sum game and consider the game is being played repeatedly. Our algorithm finds an approximate robust solution after playing the game for a number of rounds that is inversely proportional to the square of target accuracy.

1 Introduction

In most practical optimization problems the data are uncertain or imprecise due, for instance, to estimation errors or to tolerances in design implementation. The most appreciated framework to deal with such uncertainties is Robust Optimization (RO). It is a paradigm that uses ideas from convexity and duality, to construct a solution that is optimal for any realization of the uncertainty in a given set. It immunizes solutions of convex problems to bounded uncertainty in the parameters of the problem by choosing a solution that performs best against the worst possible parameter. When the objective function is convex in the parameters, and concave in the uncertainty, and when the uncertainty set is convex the overall problem is convex.

RO has several successful applications in analyzing machine learning algorithms under uncertainty. An example is solving the SVM problem when data is noisy. It has been shown [7, 23] that if the uncertainty is only in the patterns, e.g. some of its components are missing or not known precisely but the labels are known precisely, then the classification problem can be formulated as a second-order conic Program (SOCP). In [7] the underlying uncertainty set was described by multivariate normal distributions where in [23] they considered the ellipsoidal uncertainty set. However in both the cases, the problem eventually solved is more complicated than the original problem. Though we have commercially available softwares to solve SOCP, they does not scale well with the dimensionality of the problem.

In general, robust counterpart of an optimization problem is often more difficult mathematical problem. For example, with ellipsoidal uncertainties the robust counterpart of a linear program (LP) is an SOCP, where conic quadratic program becomes a semi-definite program (SDP). Both are difficult to solve than their non-robust versions. Currently we have polynomial time interior-point methods for solving SOCP/SDP. Unfortunately, these algorithms, aimed at generating high accuracy solutions, can become prohibitively time-consuming in the large-scale case. Again, with the same uncertainty assumption the robust version of SDP is NP-hard. Recently, in [2] they proposed two meta-algorithms to tackle this issue by approximately solving a robust counterpart of a given optimization problem using only a solver/oracle for the original non-robust version. They formulated the robust problem as a zero-sum game, which was solved by a primal-dual technique using tools from online convex optimization [21] , namely follow-the-regularized-leader (FTRL) and follow-the-perturbed leader

(FTPL).

Motivated from their approach, we will formulate the robust problem as a saddle-point problem and consider it as a repeated game between a forecaster (optimizer) and the environment (adversary). Our use of a game-theoretic formalism is not accidental: there exists an intimate connection between sequential prediction and some fundamental problems belonging to the theory of learning in games. We only focus on regret-based learning procedures (i.e., situations in which the players of the game base their strategies only on regrets they have suffered in the past) and our fundamental concern is whether such procedures lead to equilibria. In this work we answer this question on the affirmative, by discussing perhaps the most natural strategy for playing repeated games: *fictitious play*. It is a ”belief-based” learning rule, i.e., players form beliefs about opponent player and behave rationally with respect to these beliefs.

We show how both the players engaged in a repeated zero sum game following fictitious play gets close to their Nash equilibrium strategy after a certain number of rounds, which in turn depends upon the ”closeness” parameter (target accuracy) but independent of the dimension of the problem. Also to the best of our knowledge we are the first to establish a direct connection between Nash equilibria and robust optimal solution of the min-max problem. Along the way, we contribute some extensions to the existing fictitious play literature itself, notably extending it to the the case where both the players have infinitely many actions to choose from. Then we demonstrate applicability of our method to various practical problems ranging from machine learning to finance. We particularly discuss how our algorithm can be used to solve robust portfolio selection problem and robust classification problem.

Finally taking a slight detour we focus on the FTRL, more precisely Online Gradient Descent (OGD) based meta algorithm given in [2] and apply it to solve (approximately) the robust version of the SVM problem, invoking the original SVM oracle sufficient number of times that depends on the approximation guarantee and the size of the uncertainty set but does not directly depends on the dimension of the problem.

Organization. The rest of the work is organized as follows. In Section 2 we discuss some related approaches to solve the convex-concave saddle point problem in hand. In Section 3 we formulate the robust optimization problem more formally and describe interpretations of both saddle-point and game theoretic formulation. In Section 4 we mention the model and properties of fictitious play and see how this leads to approximate Nash equilibria of two player zero-sum games. In Section 5 we present our fictitious play based algorithm to solve the saddle-point problem and also discuss the convergence guarantees of the same. We demonstrate examples and applications of our method in Section 6. Then in Section 7 we describe the robust feasibility program and see an oracle-based approach to solve this. We also present an application of this approach in Section 7.3. We conclude in Section 8.

2 Related work

Robust optimization is by now a a field of study by itself, and the reader is referred to Ben-Tal et al. [3]; Bertsimas et al. [5] for further information, whereas Caramanis et al. [12] surveyed about applications of robust optimization in machine learning. Shalev-Shwartz’s survey [21] and Cesa-Bianchi’s book [13] are two useful references for online convex optimization and sequential prediction.

As we mentioned earlier, a significant hindrance of adopting RO to large scale problems is its increased computational complexity. This problem was addressed in several papers. In [11] they proposed to sample constraints from the uncertainty set, therefore obtain an almost robust solution with high probability drawing enough samples. The idea is similar to [2], but the main problem with their approach is that number of samples can become large as the dimension of the problem increases.

The robust classification problem itself has a vast literature. The problem of designing classifiers for uncertain observations remain an interesting open problem and has gained considerable interest in the recent past (see [24]). In Shivaswamy et al. ([22]) they used a chance constraint based approach to solve the robust SVM problem, where Lanckriet et al. ([16]) formulated the classification problem as Minimax Probabilty Machine. Both these attempts at designing robust classifiers have been limited to the case of linear classification where the uncertainty is specified over an explicitly stated feature map. Motivated by this problem Bhadra et al. ([6]) initiated a study of designing robust classifiers when the entries of the kernel matrix are independently distributed random variables. The approach, based on Chance-Constraints formalism, leads to a non-convex problem which may result in an indefinite kernel matrix.

In [1] they proposed a Robust Optimization (RO) approach which overcomes the above drawbacks. The approach employs a geometric description of uncertainty instead of the probabilistic description used earlier ([6]). They reformulated the robust counterpart as a saddle-point problem and referred to a gradient-based general scheme introduced by Nemirovski ([18]) for solving such saddle-point problems. Their work was inspired by the work of Nesterov ([19]) where a new method for minimizing a non-smooth Lipschitz continuous function f over a convex compact finite-dimensional set is proposed. The characteristic feature of Nesterov's method is that under favorable circumstances it exhibits (nearly) dimension-independent $O(1/t)$ rate of convergence, where it is assumed that the objective function f is given as a cost function of the first player in a convex-concave game. We describe a OGD based method which operates with the values and subgradients of f only, without access to the "structure" of the objective. Though our algorithm has $O(1/\sqrt{t})$ rate of convergence, to the best of our knowledge we are the first to tackle the robust optimization problem in the online learning setting.

3 Robust Optimization

Consider a general convex optimization problem

$$\min_{w\in\mathscr{W}} f(w,u) \tag{1}$$

Here f is convex in w, $\mathscr{W}\subseteq\mathbb{R}^n$ is a convex set in Euclidean space and u is a fixed parameter vector. The *robust counterpart* of (1) is given by

$$\min_{w\in\mathscr{W}} f(w,u), \forall u\in\mathscr{U} \tag{2}$$

Or equivalently

$$\min_{w\in\mathscr{W}}\max_{u\in\mathscr{U}} f(w,u) \tag{3}$$

where $\mathscr{U}$ is the uncertainty set.

3.1 Saddle-point Interpretation

The robust problem (3) can be interpreted as a saddle-point problem (see [9]) when $f(w,u)$ is concave in u for all $w\in\mathscr{W}$ and $\mathscr{U}$ is a convex set. We refer to a pair $(w^\star\in\mathscr{W}, u^\star\in\mathscr{U})$ as a saddle-point for f if

$$f(w^\star,u)\le f(w^\star,u^\star)\le f(w,u^\star)$$

for all $w\in\mathscr{W}$ and $u\in\mathscr{U}$. In other words, $w^\star$ minimizes $f(w,u^\star)$ (over $w\in\mathscr{W}$) and $u^\star$ maximizes $f(w^\star,u)$ (over $u\in\mathscr{U}$):

$$f(w^\star,u^\star)=\min_{w\in\mathscr{W}} f(w,u^\star),\ f(w^\star,u^\star)=\max_{u\in\mathscr{U}} f(w^\star,u).$$

This implies

$$\max_{u\in\mathscr{U}}\min_{w\in\mathscr{W}} f(w,u)=\min_{w\in\mathscr{W}}\max_{u\in\mathscr{U}} f(w,u) \tag{4}$$

we say that f (and $\mathscr{W}$ and $\mathscr{U}$) satisfy the *strong max-min property* or the *saddle-point property*, and that the common value is $f(w^\star,u^\star)$.

In general for any $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ (and any $\mathscr{W}\subseteq\mathbb{R}^n$ and $\mathscr{U}\subseteq\mathbb{R}^m$)

$$\max_{u\in\mathscr{U}}\min_{w\in\mathscr{W}} f(w,u)\le\min_{w\in\mathscr{W}}\max_{u\in\mathscr{U}} f(w,u) \tag{5}$$

This general inequality is called the *max-min inequality* and for convex-concave saddle-point problems equality always holds.

3.2 Game Interpretation

We can interpret the saddle-point property in terms of a continuous zero-sum game. If the first player chooses $w\in\mathscr{W}$ and the second player selects $u\in\mathscr{U}$, then player 1 pays an amount $f(w,u)$ to player 2. Player 1 therefore wants to minimize f, while player 2 wants to maximize f. (The game is called continuous since the choices are vectors, and not discrete).

If $(w^\star,u^\star)$ is a saddle-point for f, then it is called a solution of the game. $w^\star$ is called the optimal choice or strategy for player 1, and $u^\star$ is called the optimal choice or strategy for player 2. More formally, the pair $(w^\star,u^\star)$ is the Pure Strategy Nash Equilibria (PSNE) of the min-max game and $f(w^\star,u^\star)$ is the value of the game.

The *max-min inequality* (5) states the (intuitively obvious) fact that it is better for a player to go second, or more precisely, for a player to know his or her opponent's choice before choosing. In other words, the payoff to player 2 will be larger if player 1 must choose first. When the *saddle-point property* (4) holds, there is no advantage to playing second. So we will consider the players choosing actions simultaneously.

Often it is hard to compute PSNE of a game exactly. As in this case, we need to use convex-concave interior point methods, which does not scale well with the dimensionality of the problem. So, we try to solve the problem approximately.

4 ε-approximation of Two Player Zero-Sum Games

We have discussed the notion of Nash equilibria in 2-player zero-sum games. Now, We investigate whether Nash equilibria can arise as a result of the *distributed interaction* between the players of a zero-sum game, and whether the values of the players in the game are descriptive of their long-term payoffs in the course of their interaction.

Clearly, if the players are aware of the details of the game (i.e. the game's payoff matrix), they can compute their min-max strategies on the side and just use these strategies forever. We envision a much weaker distributed scenario, of *completely-uncoupled dynamics* as follows:

- each player knows her own pure strategies, but does not know the game matrix, or even the number of strategies available to her opponent;
- players interact in rounds, and each player can choose a mixed strategy in each round;
- in the end of each round, each player is informed about the expected payoff she would have gotten had she played each of her pure strategies against the opponent's mixed strategy (but the mixed strategy of the opponent is not revealed to her).

4.1 Fictitious Play

We consider a type of completely-uncoupled dynamics called fictitious play. Fictitious play was defined by George W. Brown ([10]) who conjectured its convergence to the value of a zero-sum game, and its convergence properties were established by Julia Robinson ([20]). Lets see how it works. Let $(R, C = -R)$ be a two player zero-sum game, but assume we are in a completely-uncoupled scenario where the players are ignorant of the game matrix. Informal descriptions usually depict two players playing a finite game repeatedly. After arbitrary initial moves in the first round, in every round each player plays a myopic pure best response (BR) against the empirical strategy distribution of his opponent. The following definition corresponds to the widely used version of fictitious play, where players update their beliefs *simultaneously*.

Definition 4.1. ([4]) For the two player zero-sum game $(R, C = -R)_{m\times n}$, the sequence $(i_t, j_t)_{t\in \mathbf{T}}$ is a simultaneous fictitious play (SFP) process, if $(i_1, j_1) \in m \times n$ and for all $t \in T$, $i_{t+1} \in BR_1(y_t)$ and $j_{t+1} \in BR_2(x_t)$; where the beliefs $x(t)$ and $y(t)$ are given by

$$x_t = \frac{1}{t}\sum_{s=1}^{T} e_{i_s} \text{ and } y_t = \frac{1}{t}\sum_{s=1}^{T} e_{j_s}$$

e_i is a vector whose components are all zero, except for the i^{th} component, which is 1.

4.2 Convergence of Fictitious Play

Now we will discuss the convergence properties of fictitious play.

Theorem 4.1. *([20]) If the players of a zero-sum game $(R, C = -R)$ interact via fictitious play, then:*

$$\lim_{t\to\infty} \max_i e_i^T R y_t = \lim_{t\to\infty} \min_j x_t^T R e_j = v$$

where v is the value of the row player in the game.

The statement of the Theorem 4.1 implies that the maximum payoff that the row player can achieve against the empirical strategy of the column player and the minimum loss that the column player could suffer against the empirical strategy of the row player converge to the value of the game. Now, Samuel Karlin ([15]) conjectured about the convergence speed of fictitious play.

Conjecture 4.1. Fictitious play converges with rate $\frac{1}{\sqrt{t}}$, for some function $f(|R|)$ of the description complexity of the game matrix R, i.e. for all $\varepsilon \geq 0$, for all $t \geq \frac{1}{\varepsilon^2} f^2|R|$ we have

$$|\max_i e_i^T R y_t - \min_j x_t^T R e_j| \leq \varepsilon$$

If the conjecture were true, we can establish the following convergence result of the empirical mixed strategies.

Conjecture 4.2. For all $\varepsilon \geq 0$, for all $t \geq \frac{1}{\varepsilon^2} f^2|R|$, (x_t, y_t) is an ε-approximate Nash equilibrium of the game, i.e.

1. $x_t^T R y_t \geq x'^T R y_t - \varepsilon$ for all $x' \in \Delta_m$,
2. $x_t^T C y_t \geq x_t^T C y' - \varepsilon$ for all $y' \in \Delta_n$.

That is, no player of the game can improve by more than an additive ε by switching to a different mixed strategy. Now in the next section we proceed to describe our fictitious play based meta-algorithm to solve robust optimization problem approximately.

Definition 4.2. We say the pair (w, u) is an ε-approximate solution of (3) if it satisfies the following

$$f(w, u) \leq f(w^\star, u^\star) + \varepsilon \tag{6}$$

We will see that our algorithm satisfies this bound to the class of games where each player has infinite actions.

5 Fictitious Play based Algorithm

We will now see how fictitious play can be used to solve formulation (3). The players interact in rounds as follows:

- In round $t = 1$:

  – player 1 plays an arbitrary strategy $w_1 \in \mathscr{W}$ and player 2 plays an arbitrary strategy$u_1 \in \mathscr{U}$

  – player 1 observes loss $f(w, u_1)$ (convex in w) and player 2 observes gain $f(w_1, u)$ (concave in u).

- In round $t = 2$:

  – player 1 plays any strategy $w_2 \in argmin_{w \in \mathscr{W}} f(w, u_1)$ and player 2 plays any strategy $u_2 \in argmax_{u \in \mathscr{U}} f(w_1, u)$

  – player 1 observes loss $f(w, u_2)$ (convex in w) and player 2 observes gain $f(w_2, u)$ (concave in u).

- In a general round t:

  – player 1 plays any strategy

  $$w_t \in argmin_{w \in \mathscr{W}} [\frac{1}{t-1} \sum_{k=1}^{t-1} f(w, u_k)]$$

  and player 2 plays any strategy

  $$u_t \in argmax_{u \in \mathscr{U}} [\frac{1}{t-1} \sum_{k=1}^{t-1} f(w_k, u)]$$

  – player 1 observes $f(w, u_t)$ (convex in w) and player 2 observes $f(w_t, u)$ (concave in u)

Observe that fictitious play can be viewed equivalently as the result of the two-players of a zero-sum game using the "Follow-the-Leader" (FTL) protocol to update their strategies. Here player 1 tries to minimize his cumulative loss over time and player 2 tries to maximize his cumulative gain over time. In online learning community (see [21])it is well known that the performance of FTL can be very poor and a widely used solution is is to incorporate a regularizer term, which is known as "Follow-the-Regularized-Leader" (FTRL):

$$w_t \in argmin_{w \in \mathscr{W}} [\frac{1}{t-1} \sum_{k=1}^{t-1} f(w, u_k) + R(w)] \tag{7}$$

$$u_t \in argmax_{u \in \mathscr{U}} [\frac{1}{t-1} \sum_{k=1}^{t-1} f(w_k, u) + R(u)] \tag{8}$$

It is also quite well known that using squared norm regularizer in (7) and (8), that is using $R(x) = \frac{\|x\|_2^2}{2\eta}$, we will get Online-Gradient-Descent updates for w and u. Before stating our OGD based algorithm formally, we first make the following assumptions:

1. The feasible sets $\mathscr{W}$ and $\mathscr{U}$ are bounded, closed and non-empty.

2. For all t, $f(w, u_t)$ and $f(w_t, u)$ is differentiable w.r.t w and u respectively.

3. For all t, there exists an algorithm, given w and u, which produces $\nabla_w f(w, u_t)$ and $\nabla_u f(w_t, u)$.

4. For all $v \in {}^n$, there exists an algorithm which can produce $argmin_{w \in \mathscr{W}} \|w - v\|_2$ and $argmin_{u \in \mathscr{U}} \|u - v\|_2$. We define the projections $\prod_{\mathscr{W}}(v) = argmin_{w \in \mathscr{W}} \|w - v\|_2$ and $\prod_{\mathscr{U}}(v) = argmin_{u \in \mathscr{U}} \|u - v\|_2$.

Also in the setting of this section, we shall make use of the following definitions.

- D_w and D_u denote l_2 diameters of $\mathscr{W}$ and $\mathscr{U}$ respectively, that is $D_w = \max_{x,y \in \mathscr{W}} \|x - y\|_2$ and $D_u = \max_{x,y \in \mathscr{U}} \|x - y\|_2$.

- $G_w \geq \|\nabla_w f(w, u)\|_2$ and $G_u \geq \|\nabla_u f(w, u)\|_2$ for all w, u are upper bounds of the gradients of f.

- $\eta_w = \frac{D_w}{G_w \sqrt{T}}$ and $\eta_u = \frac{D_u}{G_u \sqrt{T}}$ be the learning rate for w and u respectively.

With the above assumptions and definitions, we can now present a meta algorithm for robust optimization, given in Algorithm 1, which is comprised of primal-dual iterations.

Algorithm 1

```
Input: parameters $D_w$, $D_u$, $G_w$, $G_u$, target accuracy $\varepsilon > 0$
Output: $\varepsilon$-approximate solution to (3)
set $T = \lceil (\frac{G_u D_u + G_w D_w}{\varepsilon})^2 \rceil$, $\eta_w = \frac{D_w}{G_w \sqrt{T}}$, $\eta_u = \frac{D_u}{G_u \sqrt{T}}$
initialize $w \in \mathscr{W}$ and $u \in \mathscr{U}$ arbitrarily
for $t = 2$ to $T$ do
    update $w_t \leftarrow \prod_{\mathscr{W}} [w_{t-1} - \eta_w \nabla_w f(w_{t-1}, u_{t-1})]$
    update $u_t \leftarrow \prod_{\mathscr{U}} [u_{t-1} - \eta_u \nabla_u f(w_{t-1}, u_{t-1})]$
end for
return $\overline{w}_T = \frac{1}{T} \sum_{t=1}^{T} w_t$, $\overline{u}_T = \frac{1}{T} \sum_{t=1}^{T} u_t$
```

5.1 Convergence Analysis

In this section we will restate the convergence results from section 4.2 to the class of of two player zero-sum games where each player has infinitely many actions, as we have seen in Algorithm 1. For this algorithm, we prove:

Theorem 5.1. *For all $\varepsilon \geq 0$, for all $T \geq (\frac{G_u D_u + G_w D_w}{\varepsilon})^2$ we have*

$$\max_{u \in \mathscr{U}} f(\overline{w}_T, u) - \min_{w \in \mathscr{W}} f(w, \overline{u}_T) \leq \varepsilon$$

Proof: Considering Regret Guarantee of Online Gradient Descent [25], we have

$$\frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) - \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) \le \frac{G_w D_w}{\sqrt{T}} \tag{9}$$

$$\max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) - \frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) \le \frac{G_u D_u}{\sqrt{T}} \tag{10}$$

From (9) and (10) we get

$$\max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) - \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) \le \frac{G_w D_w}{\sqrt{T}} + \frac{G_u D_u}{\sqrt{T}}$$

As $T \ge (\frac{G_u D_u + G_w D_w}{\varepsilon})^2$ we have

$$\max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) \le \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) + \varepsilon$$

Now, as f is convex w.r.t w and concave w.r.t u,

$$f(\overline{w}_T,u) \le \frac{1}{T}\sum_{t=1}^{T} f(w_t,u),$$

$$f(w,\overline{u}_T) \ge \frac{1}{T}\sum_{t=1}^{T} f(w,u_t)$$

Using these facts in the above the result follows. □

Remark 5.1. Notice that by proving Theorem 5.1 we also provide a formal proof of Conjecture 4.1 for zero-sum game with infinite actions.

Now we will show that output of our algorithm is indeed the ε-approximate Nash equilibria of the game.

Theorem 5.2. *For all $\varepsilon \ge 0$, for all $T \ge (\frac{G_u D_u + G_w D_w}{\varepsilon})^2$, $(\overline{w}_T,\overline{u}_T)$ is ε-approximate PSNE of the game.*

Proof: From Theorem 5.1 we have

$$\max_{u\in\mathscr{U}} f(\overline{w}_T,u) \le \min_{w\in\mathscr{W}} f(w,\overline{u}_T) + \varepsilon$$

$$\Rightarrow f(\overline{w}_T,u) \le \min_{w} f(w,\overline{u}_T) + \varepsilon, \text{ for all } u \in \mathscr{U}$$

$$\Rightarrow f(\overline{w}_T,u) \le f(\overline{w}_T,\overline{u}_T) + \varepsilon, \text{ for all } u \in \mathscr{U}$$

$$\Rightarrow f(\overline{w}_T,\overline{u}_T) \ge f(\overline{w}_T,u) - \varepsilon, \text{ for all } u \in \mathscr{U} \tag{i}$$

Again,

$$\min_{w\in\mathscr{W}} f(w,\overline{u}_T) \ge \max_{u\in\mathscr{U}} f(\overline{w}_T,u) - \varepsilon$$

$$\Rightarrow f(w,\overline{u}_T) \ge \max_{u\in\mathscr{U}} f(\overline{w}_T,u) - \varepsilon, \text{ for all } w \in \mathscr{W}$$

$$\Rightarrow f(w,\overline{u}_T) \ge f(\overline{w}_T,\overline{u}_T) - \varepsilon, \text{ for all } w \in \mathscr{W}$$

$$\Rightarrow f(\overline{w}_T,\overline{u}_T) \le f(w,\overline{u}_T) + \varepsilon, \text{ for all } w \in \mathscr{W} \tag{ii}$$

From (i) and (ii), we conclude that $(\overline{w}_T,\overline{u}_T)$ is indeed an ε-approximate PSNE of the game. □

Remark 5.2. See we have also established the result of Conjecture 4.2.

The following theorem shows that the output of Algorithm 1 indeed converges to the saddle-point or equivalently PSNE of the game.

Theorem 5.3. *For all $\varepsilon \ge 0$, ε-approximate PSNE profile $(\overline{w}_T,\overline{u}_T)$ converges to exact PSNE profile $(w^\star,u^\star)$ as $T \to \infty$*

Proof: Before proving Theorem 5.3 formally, we first claim the following:

Claim 5.1. $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) = f(w^\star,u^\star)$

Proof: From (9) and (10) we have

$$\lim_{T\to\infty} \max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) \le \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) \le \lim_{T\to\infty} \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) \tag{11}$$

Again, $\min_{w} \sum_{t=1}^{T} f(w,u_t) \le \sum_{t=1}^{T} f(w_t,u_t) \le \max_{u} \sum_{t=1}^{T} f(w_t,u)$

Dividing by T and taking limit we get

$$\lim_{T\to\infty} \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) \le \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) \le \lim_{T\to\infty} \max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) \tag{12}$$

(11) and (12) implies,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u_t) = \lim_{T\to\infty} \max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) = \lim_{T\to\infty} \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) \tag{13}$$

Now using the similar arguments as [20] and Theorem 4.1, we can state that

$$\lim_{T\to\infty} \max_{u} \frac{1}{T}\sum_{t=1}^{T} f(w_t,u) = \lim_{T\to\infty} \min_{w} \frac{1}{T}\sum_{t=1}^{T} f(w,u_t) = f(w^\star,u^\star) \tag{14}$$

From (13) and (14), the result follows. □

Remark 5.3. Observe that (14) basically is a restatement of Robinson's result (Theorem 4.1), specially modified for our setting. Also notice that Claim 5.1 implies long-term average payoff for both the players converges to the value of the game.

Proof of Theorem 5.3: Using convexity and concavity property of f, we have

$$f(\overline{w}_T, \overline{u}_T) \leq \max_u f(\overline{w}_T, u) \leq \max_u \frac{1}{T} \sum_{t=1}^{T} f(w_t, u) \text{ and}$$

$$f(\overline{w}_T, \overline{u}_T) \geq \min_w f(w, \overline{u}_T) \geq \min_w \frac{1}{T} \sum_{t=1}^{T} f(w, u_t)$$

This implies,

$$\min_w \frac{1}{T} \sum_{t=1}^{T} f(w, u_t) \leq f(\overline{w}_T, \overline{u}_T) \leq \max_u \frac{1}{T} \sum_{t=1}^{T} f(w_t, u) \quad (15)$$

Now consider the following theorem.

Theorem 5.4. ***(Sandwich Theorem)*** *Let $D \subset \mathbb{R}$ and f, g, h be functions on D to $\mathbb{R}$. Let $c \in D'$.*
If $f(x) \leq g(x) \leq h(x)$ for all $x \in D - \{c\}$ and if $\lim_{x \to c} f(x) = \lim_{x \to c} h(x) = l$, then $\lim_{x \to c} g(x) = l$

Using Theorem 5.4 on (15) with T as the variable and from (14) we have

$$\lim_{T \to \infty} f(\overline{w}_T, \overline{u}_T) = f(w^\star, u^\star)$$

As f is continuous in both of its arguments,

$$f(\lim_{T \to \infty} \overline{w}_T, \lim_{T \to \infty} \overline{u}_T) = \lim_{T \to \infty} f(\overline{w}_T, \overline{u}_T)$$

$$\Rightarrow f(\lim_{T \to \infty} \overline{w}_T, \lim_{T \to \infty} \overline{u}_T) = f(w^\star, u^\star)$$

Now, if $(w^\star, u^\star)$ is unique, then

$$w^\star = \lim_{T \to \infty} \overline{w}_T, \; u^\star = \lim_{T \to \infty} \overline{u}_T$$

thereby ensuring unique convergence. But if $(w^\star, u^\star)$ is not unique, that is if multiple Nash equilibria exists, then $(\overline{w}_T, \overline{u}_T)$ converge to any one of the PSNE's, $(w^\star, u^\star)$ being one of them. □

Now we will establish a connection between Nash equilibria and robust solution of two player zero-sum games. We will state it as a corollary of Theorem 5.1.

Corollary 5.1. For all $\varepsilon \geq 0$, for all $T \geq (\frac{G_u D_u + G_w D_w}{\varepsilon})^2$, ε-approximate Nash Equilibrium profile is the ε-approximate robust solution to the min-max problem.

Proof: From Theorem 5.1,

$$\max_{u \in \mathscr{U}} f(\overline{w}_T, u) \leq \min_w f(w, \overline{u}_T) + \varepsilon$$

$$\Rightarrow \max_{u \in \mathscr{U}} f(\overline{w}_T, u) \leq f(w^\star, \overline{u}_T) + \varepsilon$$

$$\Rightarrow \max_{u \in \mathscr{U}} f(\overline{w}_T, u) \leq f(w^\star, u^\star) + \varepsilon$$

$$\Rightarrow f(\overline{w}_T, u) \leq f(w^\star, u^\star) + \varepsilon, \text{ for all } u \in \mathscr{U} \quad \text{(iii)}$$

$$\Rightarrow f(\overline{w}_T, \overline{u}_T) \leq f(w^\star, u^\star) + \varepsilon$$

So function value at ε-approximate Nash Equilibrium profile is ε-close to the value of the game, satisfying Definition 4.2. □

Remark 5.4. Also from (iii) we Observe that for all $\varepsilon \geq 0$, for all $T \geq (\frac{G_u D_u + G_w D_w}{\varepsilon})^2$

$$f(\overline{w}_T, u^\star) \leq f(w^\star, u^\star) + \varepsilon,$$

that is under worst possible noise, value of f at $\overline{w}_T$ is at most ε-worse than value of f at $w^\star$, which is the exact robust solution of the problem. This implies $\overline{w}_T$ is ε-approximate robust solution.

6 Applications

In this section we will show one example from finance and another one from machine learning to indicate vast usefulness of our results.

6.1 Robust Classification Problem Under Uncertainty in Kernel Matrices

Given a set of training data $\{(x_i, y_i) | y_i \in \pm 1\}$, the robust SVM problem with uncertainty in kernel matrix can be cast as follows (see [1])

$$\max_{\alpha \in S_n} \min_{K \in E(k)} -\frac{1}{2} \alpha^T Y K Y \alpha + \alpha^T e \quad (16)$$

where $S_n = \{\alpha | 0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0\}$, $Y = diag(y_1, ..., y_n)$, e is a vector of all 1's. Each entry of the matrix K, is defined by $K_{ij} = K(xi, xj)$ where K is a kernel function and defines a dot product in an associated *Reproducing Kernel Hilbert Space*, thus needs to be positive semi-definite. The uncertainty in the kernel matrix K is modeled by a bounded convex set $E(k)$, which encompasses several possible realizations of K

$$E(k) = \{K = \overline{K} + \sum_{l=1}^{L} \eta_l K_l, \|\eta\|_2 \leq k, \eta_l \geq 0, l = 1, 2, ..., L\} \quad (17)$$

The constraint $\eta_l \geq 0$ is needed to ensure that each element in the set represents a valid kernel evaluation. The quantity k measures the quality of approximation and hence the uncertainty. If $k = 0$ then we have no uncertainty. As

k increases the uncertainty set increases. The matrices $K, K_l \in S^n_+$ are obtained by evaluating the known kernel functions K, K_l on the training set. As any $K \in E(k)$ is always positive semi-definite, the set $E(k)$ defines a valid model for describing uncertainty in psd matrices.

The Robust SVM problem (16) with uncertain K, as characterized in (17), can now be cast as follows (see [1])

$$\max_{\alpha \in S_n} \min_{\|\eta\|_2 \leq k} -\frac{1}{2}\alpha^T Y \overline{K} Y \alpha - \frac{1}{2}\alpha^T Y \sum_{l=1}^{L} \eta_l K_l Y \alpha + \alpha^T e \tag{18}$$

Now (18) can be cast as a Conic Quadratic problem. Such problems can be solved in polynomial time by Interior Point (IP) algorithm. However for large-scale problems IP methods become intractable. Now we will show that this is a saddle-point problem and algorithm 1 can be used to learn the robust classifier. for the sake of convenience we rename the variables, in particular we use $\eta \to x, \alpha \to y, y \to s, YK_lY \to Q_l, Y\overline{K}Y \to \overline{Q}, k \to 1$ to reformulate (18) as the following

$$\max_{y \in \mathscr{Y}} \min_{x \in \mathscr{X}} -\frac{1}{2} y^T \overline{Q} y - \sum_{l=1}^{L} x_l (\frac{1}{2} y^T Q_l y) + y^T e \tag{19}$$

Where $\mathscr{Y} = \{y \in \mathbf{R}^n : 0 \leq y_i \leq C, \sum\limits_{i=1}^{n} s_i y_i = 0\}$ and $\mathscr{X} = \{x \in \mathbf{R}^l : x \geq 0, \|x\|_2 \leq 1\}$

Here the objective function $f(x,y)$ is linear in x and concave in y, as Hessian, defined by $\nabla^2_y f(x,y) = -(\overline{Q} + \sum\limits_l x_l Q_l)$, is a negative linear combination of p.s.d matrices. Also it is easy to verify that both $\mathscr{X}$ and $\mathscr{Y}$ are convex, closed and bounded. So, the robust formulation (19) is amenable to Algorithm 1. In this case we have

$$\nabla_x f = -d \; ; d = [d_1, ..., d_L]^T, d_l = \frac{1}{2} y^T Q_l y \text{ and}$$
$$\nabla_y f = -\overline{Q} y - \sum_{l=1}^{L} x_l Q_l y + e,$$

so that in each iteration of the algorithm, the update of the variables takes the simple form

$$x_t = \prod\nolimits_{\mathscr{X}} [x_{t-1} + \eta_x d_{t-1}]; \; d_{l,t-1} = \frac{1}{2} y^T Q_l y$$
$$y_t = \prod\nolimits_{\mathscr{Y}} [y_{t-1} + \eta_y(-\overline{Q} y_{t-1} - \sum_{l=1}^{L} x_{l,t-1} Q_l y_{t-1} + e)]$$

where η_x, η_y are the learning rates for x and y respectively. Now,

$$\mathbf{D_x} = \max_{u,v \in \mathscr{X}} \|u - v\|_2 = \sqrt{\mathbf{2}},$$
$$\mathbf{D_y} = \max_{u,v \in \mathscr{Y}} \|u - v\|_2 = \mathbf{C}\sqrt{\mathbf{n}}$$

are diameter of the sets $\mathscr{X}$ and $\mathscr{Y}$ respectively and

$$\|\nabla_\mathbf{x} \mathbf{f}\|_\mathbf{2} = \sqrt{d_1^2 + ... + d_L^2} \leq \sqrt{L} \max_l d_l \leq \frac{\sqrt{L}}{2} \max_l y^T Q_l y \leq \frac{\sqrt{L}}{2} \max_l (\lambda_{\max}(Q_l) y^T y) \leq \frac{C^2 n \sqrt{L}}{2} \max_l (\lambda_{\max}(Q_l)) = \mathbf{G_x},$$

$$\|\nabla_\mathbf{y} \mathbf{f}\|_\mathbf{2} = \|\overline{Q} y\|_2 + \| \sum_{l=1}^{L} x_l Q_l y\|_2 + \|e\|_2 \leq \|\overline{Q}\|_2 \|y\|_2 + \sum_{l=1}^{L} x_l \|Q_l\|_2 \|y\|_2 + \|e\|_2 \leq \sqrt{n}(1 + C(\|\overline{Q}\|_2 + \max_l \|Q_l\|_2)) = \mathbf{G_y}$$

are upper bound of the gradients of f.

6.2 Robust Portfolio Optimization

The classical work of Markowitz ([17]) served as the genesis for modern portfolio theory. The canonical problem is to allocate wealth across n risky assets with mean returns $\mu \in \mathbb{R}^n$ and return covariance matrix $Q \in \mathbb{S}^n_+$ over a weight vector $x \in \mathbb{R}^n$. This can be done in three essentially equivalent ways: (i) maximize expected return subject to an upper limit on the variance, (ii) minimize the variance subject to a lower limit on the expected return, (iii) maximize the risk-adjusted expected return. These three problems are parametrized by the variance limit, expected return limit, and the risk-aversion parameter, respectively. Here we will focus on the *risk-adjusted return* formulation:

$$\max_{x \in \mathscr{X}} \mu^T x - \lambda x^T Q x \tag{20}$$

Above μ_i, the ith component of the vector μ, denotes the estimated expected return of security i. Diagonal elements q_{ii} of the Q matrix denote the variance of the return on security i while off-diagonal elements q_{ij} denote the covariance between the returns of securities i and j. The components x_i of the variable vector x denote the proportion of the portfolio to be invested in security i. The scalar λ is the risk aversion parameter and $\mathscr{X}$ represents the set of acceptable weight vectors ($\mathscr{X}$ typically contains the normalization constraint $e^T x = 1$ and often has no short-sales constraints, i.e., $x_i \geq 0$, $i = 1, ..., n$, among others).

Despite the widespread popularity of this approach, a fundamental drawback from the practitioner's perspective is that μ and Q are rarely known with complete precision. Robust models for the mean and covariance information are a natural way to alleviate this difficulty, and they have been explored by numerous researchers. Here we will consider the uncertainty model used by Goldfarb and Iyengar ([14]). The mean return vector μ and return covariance matrix Q is assumed to lie in respective uncertainty sets S_μ and S_Q given by

$$S_\mu = \{\mu : \mu = \mu_0 + \xi, |\xi_i| \leq \gamma_i\}$$

$$S_Q = \{Q : Q = Q_0 + W, \|W\|_F \leq \rho\}$$

Here, μ_0 and Q_0 are fixed and γ and ρ denote the level of uncertainty respectively for μ and Q. Now As pointed out before, the primary criticism leveled against the Markowitz model is that the optimal portfolio is extremely sensitive to the market parameters, since these parameters are estimated from noisy data. By introducing measures of uncertainty in the market models, we are attempting to correct this sensitivity to perturbations. The uncertainty sets S_μ and $\S_Q$ represent the uncertainty of our limited (inexact) information of the market parameters, and we wish to select portfolios that perform well for all parameter values that are consistent with this limited information. Such portfolios are solutions of appropriately defined min-max optimization problems called *robust portfolio selection problem*

$$\max_{x \in \mathscr{X}} \min_{\mu \in S_\mu, Q \in S_Q} \mu^T x - \lambda x^T Q x \tag{21}$$

Now, in [14] they showed (21) can be reformulated as an SOCP, which gets very difficult to solve for large scale problems. Now we will show that this is a saddle-point problem and our algorithm can be used to compute robust efficient frontiers. Here, the objective function is

$$f(x, \mu, Q) = \mu^T x - \lambda x^T Q x$$

which is linear in μ and Q and concave in x (as Q is a positive semi-definite matrix). Also it is easy to verify that $\mathscr{X}, S_\mu, S_Q$ all are convex, closed and bounded. So, the robust quadratic program (21) is amenable to Algorithm 1. In this case we have

$$\nabla_x f = \mu - 2\lambda Q x,$$
$$\nabla_\mu f = x \text{ and}$$
$$\nabla_Q f = -\lambda x x^T,$$

so that in each iteration of the algorithm, the update of the variables takes the simple form

$$x_t = \textstyle\prod_{\mathscr{X}}[x_{t-1} + \eta_x(\mu_{t-1} - 2\lambda Q_{t-1} x_{t-1})]$$
$$\mu_t = \textstyle\prod_{S_\mu}[\mu_{t-1} - \eta_\mu x_{t-1}]$$
$$Q_t = \textstyle\prod_{S_Q}[Q_{t-1} + \eta_Q \lambda x_{t-1} x_{t-1}^T]$$

where η_x, η_μ, η_Q are the learning rates for x, μ, Q respectively. Now,

$$\mathbf{D_x} = \max_{u,v \in \mathscr{X}} \|u - v\|_2 = \sqrt{\mathbf{2}},$$

$$\mathbf{D}_\mu = \max_{\mu_1, \mu_2 \in S_\mu} \|\mu_1 - \mu_2\|_2 = \max \|\xi_1 - \xi_2\|_2 \leq \max(\|\xi_1\|_2 + \|\xi_2\|_2) = \mathbf{2}\|\gamma\|_\mathbf{2} \text{ and}$$

$$\mathbf{D_Q} = \max_{Q_1, Q_2 \in S_Q} \|Q_1 - Q_2\|_2 \leq \max(\|Q_1\|_2 + \|Q_2\|_2) \leq \max(\|Q_1\|_F + \|Q_2\|_F) = \mathbf{2}\rho$$

are diameter of the sets $\mathscr{X}, S_\mu, S_Q$ respectively and

$$\|\nabla_\mathbf{x} \mathbf{f}\|_\mathbf{2} = \|\mu - 2\lambda Q x\|_2 \leq \|\mu\|_2 + 2\lambda\|Qx\|_2 \leq \|\mu\|_2 + 2\lambda\|Q\|_2\|x\|_2 \leq \|\mu\|_2 + 2\lambda\|Q\|_2 \leq (\|\mu_0\|_2 + \|\xi\|_2) + 2\lambda(\|Q_0\|_2 + \|W\|_2) \leq (\|\mu_0\|_2 + \|\gamma\|_2) + 2\lambda(\|Q_0\|_2 + \rho) = \mathbf{G_x},$$

$$\|\nabla_\mu \mathbf{f}\|_\mathbf{2} = \|x\|_2 \leq 1 = \mathbf{G}_\mu \text{ and}$$

$$\|\nabla_\mathbf{Q} \mathbf{f}\|_\mathbf{2} = \lambda\|xx^T\|_2 = \lambda\|xx^T\|_F = \lambda Trace(xx^T) = \lambda\|x\|_2^2 \leq \lambda = \mathbf{G_Q}$$

are upper bound of the gradients of f.

7 Robust Feasibility Problem : A Detour

In this section we switch our attention to feasibility problems and its robust counterparts. We present an oracle based algorithm to solve general robust feasibility problems approximately and see its application in classification problems, namely support vector machines (SVM).
Consider a general convex feasibility problem:

$$\exists ? x \in \mathscr{D} : f_i(x, u_i) \leq 0, \forall i \in [m]. \tag{22}$$

Here $f_1, \ldots, f_m$ are convex functions in x, $\mathscr{D} \subseteq \mathbb{R}^n$ is a convex set in Euclidean space and $u_1, \ldots, u_m$ are fixed parameter vectors. The **robust counterpart** of (22) is given by:

$$\exists ? x \in \mathscr{D} : f_i(x, u_i) \leq 0, \forall u_i \in \mathscr{U}, \forall i \in [m]. \tag{23}$$

As in [2], we say $x \in \mathscr{D}$ is an *ε-approximate solution* to this problem if x meets each constraint up to ε, that is, it satisfies

$$f_i(x, u_i) \leq \varepsilon, \forall u_i \in \mathscr{U}, \forall i \in [m].$$

7.1 Oracle-Based Robust Optimization

In [2], they assumed that there exists an oracle or solver for the original optimization problem (22). This oracle approximately solves formulation (23) for any fixed noise vectors $u_1, \ldots, u_m$: $u_i \in \mathscr{U}$, $\forall i \in [m]$. It either returns an ε-feasible solution, that is, it returns a vector $x \in \mathscr{D}$ that satisfies

$$f_i(x,u_i) \leq \varepsilon, \forall i \in [m]$$

or declares that the problem is infeasible if $\nexists x \in \mathscr{D}$ for which

$$f_i(x,u_i) \leq 0, \forall i \in [m].$$

7.2 Online Gradient Descent based Algorithm

The robust problem (23) can be formulated as a convex-concave saddle-point problem by making the following assumptions:

1. $f_i(x,u)$ concave in u for all $x \in \mathscr{D}$, for all $i \in [m]$.
2. The uncertainty set $\mathscr{U}$ is convex.

Now define:

- $D \geq max_{u,v \in \mathscr{U}} \|u-v\|_2$, the l_2 diameter of the uncertainty set.
- $\| \bigtriangledown_u f_i(x,u)\|_2 \leq G$, the upper bound over the gradients, for all $x \in \mathscr{D}$ and $u \in \mathscr{U}$.

With these assumptions and definitions, in [2] they have given a meta-algorithm, which is comprised of primal-dual iterations, where the dual part updates the noise terms according to the current primal solution, via an online gradient ascent based update [25].

The approximate robust solution is obtained by invoking the oracle a finite number of times, where the number of iterations is a function of G, D, ε. At each iteration, in the dual step, the current noise vectors are updated using gradient ascent (projected) rule:

$$u_i^t \leftarrow \prod\nolimits_{\mathscr{U}} [u_i^{t-1} + \eta \bigtriangledown_u f_i(x^{t-1}, u_i^{t-1})], \forall i \in [m]$$

Next in the primal step, the oracle is called using current noise samples to obtain the current solution x^t. Finally, the algorithm returns either simple average of the primal solutions obtained in each iteration or declare the problem is infeasible if the original oracle does so at any iteration.

7.3 Application: Oracle Based Robust SVM

Consider the standard hard margin SVM problem:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|_2^2 \\ &\text{subject to } y_i(w^T x_i + b) \geq 1, \forall i \in [m] \end{aligned} \tag{24}$$

The robust counterpart of this optimization problem is a second-order conic program (SOCP) that can be solved in polynomial time using interior point methods. However, recall that the goal is to solve the robust problem by invoking a solver of the original (non-robust) optimization problem. In the discussion below, we will assume that the uncertainty set $\mathscr{U}$ is the Euclidean unit ball, that is

$$\mathscr{U} = \{u \in \mathbb{R}^d : \|u\|_2 \leq 1\}.$$

Now assume that x takes values in an ellipsoid with center $\bar{x}$, metric Σ and radius γ, that is

$$x \in \mathscr{B} := \{x : (x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2.$$

It also implies that $x = \bar{x} + \gamma \Sigma^{1/2} u$, where $u \in \mathscr{U}$. The robustness criteria can be enforced by requiring that we classify x correctly for all $x \in \mathscr{B}(\bar{x}, \Sigma, \gamma)$, that is

$$y\{w^T(\bar{x} + \gamma \Sigma^{1/2} u) + b\} \geq 1, \text{ for all } u \in \mathscr{U}.$$

So, the robust SVM problem can be written along the line of (22) as,

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|_2^2 \\ &\text{subject to } 1 - y_i\{w^T(\bar{x}_i + \gamma_i \Sigma_i^{1/2} u_i) + b\} \leq 0, \\ &\qquad \forall u_i \in \mathscr{U}, \forall i \in [m] \end{aligned} \tag{25}$$

This is equivalent to the SOCP formulation [22] :

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|_2^2 \\ &\text{subject to } y_i(w^T \bar{x}_i + b) \geq 1 + \gamma_i \|\Sigma_i^{1/2} w\|, \forall i \in [m] \end{aligned} \tag{26}$$

7.4 OGD-based Robust SVM algorithm

The robust SVM program (25) is amenable to the OGD-based meta-algorithm (Algorithm $\mathscr{A}$), as in this case, the constraints are of the form:

$$f(w,u) = 1 - y\{w^T(\bar{x} + \gamma \Sigma^{1/2} u) + b\}.$$

The constraints are linear with respect to the the noise term u and the uncertainty set, Euclidean unit ball, is convex. Therefore satisfies all the assumptions of Section 7.3. In this case we have $\bigtriangledown_u f(w,u) = -y\gamma\Sigma^{1/2}u$.

Now we will see the oracle based robust SVM algorithm.

Algorithm 2 OGD-based SVM

```
**Input:** Tuples $(\bar{x}_i, \Sigma_i, \gamma_i)$, $\forall i \in [m]$, target accuracy $\varepsilon > 0$
**Output:** $2\varepsilon$-approximate solution to (25), or *infeasibe*
set $D = 2$, $G = \sqrt{max_{i=1}^{m}(\gamma_i^2 \lambda_{max}(\Sigma_i))}$
set $T = \lceil \frac{G^2 D^2}{\varepsilon^2} \rceil$ and $\eta = \frac{D}{G\sqrt{T}}$
initialize $(u_1^0, ..., u_m^0)$ and $w_0$ arbitrarily
**for** $t = 1, 2, ..., T$ **do**
  **for** $i = 1, 2, ..., m$ **do**
    update $u_i^t \leftarrow \frac{u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}}{max\{\|u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}\|_2, 1\}}$
    compute $x_i^t = \bar{x}_i + \gamma_i \Sigma_i^{1/2} u_i^t$
  **end for**
  set $(w_t, b_t) \leftarrow \mathscr{O}_\varepsilon(x_1^t, ..., x_m^t)$
  **if** oracle declared infeasibility **then return** *infeasible*
  **else** set $w_t \leftarrow \frac{w_t}{max\{\|w_t\|_2, 1\}}$
**end for**
**return** $\overline{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$, $\overline{b} = \frac{1}{T}\sum_{t=1}^{T} b_t$
```

Here the optimization oracle ($\mathscr{O}_\varepsilon$) is the original non-robust ε-approximate SVM solver, which is a quadratic programming problem and thus it avoids solving the SOCP formulation.

```
**Oracle** $\mathscr{O}_\varepsilon$

**Input:** noise vectors $x_1, ..., x_m$, $x_i \in \mathscr{B}(\bar{x}_i, \Sigma_i, \gamma_i)$, $\forall i \in [m]$
**Output:** vector $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ which solves:
    minimize $\frac{1}{2}\|w\|^2$
    subject to $y_i(w^T x_i + b) \geq 1 - \varepsilon$, $\forall i \in [m]$

or *infeasible* if $\nexists w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ for which
    $y_i(w^T x_i + b) \geq 1$, $\forall i \in [m]$
```

For Algorithm 2, we prove:

Theorem 7.1. *Algorithm 2 returns an 2ε-approximate robust solution, to (25), that is $y_i(\overline{w}^T x_i + \overline{b}) \geq 1 - 2\varepsilon$, $\forall i \in [m]$, after at most $T = O(G^2/\varepsilon^2)$ calls to the SVM-oracle, where $G = \sqrt{max_{i=1}^{m}(\gamma_i^2 \lambda_{max}(\Sigma_i))}$*

Proof: Note that for all $u_i \in \mathscr{U}$, the d-dimensional unit ball, and $\|w\|_2 \leq 1$ we have

$$max_{u,v \in \mathscr{U}} \|u - v\|_2 = 2 \text{ and}$$

$$\| \bigtriangledown_u f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{max}(\Sigma_i)$$

where λ_{max} denotes the maximum eigenvalue.
Now suppose a solution $(\overline{w}, \overline{b})$ is returned by Algorithm 2. This implies

$$y_i(w_t^T x_i^t + b_t) \geq 1 - \varepsilon \text{ for all } t \in [T] \text{ and } i \in [m], \text{ i.e.}$$

$$\text{for all } i \in [m],\ \frac{1}{T}\sum_{t=1}^{T} y_i(w_t^T x_i^t + b_t) \geq 1 - \varepsilon \qquad (27)$$

Now from the regret guarantee of OGD [25],

$$max_{x_i} \frac{1}{T}\sum_{t=1}^{T}[1 - y_i(w_t^T x_i + b_t)] - \frac{1}{T}\sum_{t=1}^{T}[1 - y_i(w_t^T x_i^t + b_t)] \leq \frac{GD}{\sqrt{T}} \leq \varepsilon \qquad (28)$$

Combining (27) and (28) we have,

$$\varepsilon \geq \frac{1}{T}\sum_{t=1}^{T}[1 - y_i(w_t^T x_i^t + b_t)] \geq max_{x_i} \frac{1}{T}\sum_{t=1}^{T}[1 - y_i(w_t^T x_i + b_t)] - \varepsilon = max_{x_i}[1 - y_i(\overline{w}^T x_i + \overline{b})] - \varepsilon$$

where the final equality follows from the linearity of the constraints with respect to w and b. Hence, we have

$$y_i(\overline{w}^T x_i + \overline{b}) \geq 1 - 2\varepsilon,\ \forall x_i \in \mathscr{B}(\bar{x}_i, \overline{\Sigma}_i, \gamma_i),\ \forall i \in [m].$$

implying that $(\overline{w}, \overline{b})$ is a 2ε-approximate robust solution. Now Setting $D = 2$ and $G = \sqrt{max_{i=1}^{m}(\gamma_i^2 \lambda_{max}(\Sigma_i))}$, we obtain the result. □

Similarly we can extend Algorithm 2 and Theorem 7.1 to the soft-margin SVM problem also.

7.5 Experiments

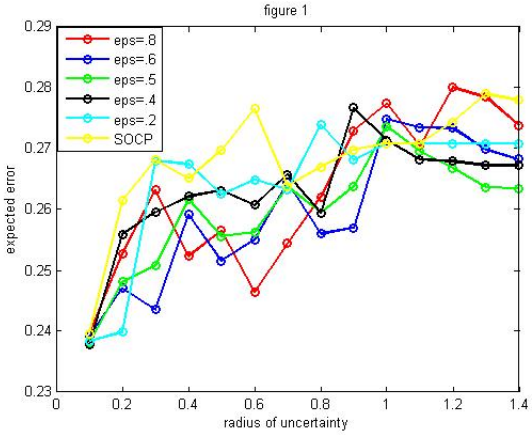
We denote formulation (26) as SVM-SOCP and Algorithm 2 as SVM-OGD. Now we introduce two error measures [22].

Worst case error ($\mathbf{e_{wc}}$): Let $x \in \mathscr{B}(\bar{x}, \Sigma, \gamma)$ has true label y. For this ellipsoid the worst case error is given by

$$e_{wc}(\mathscr{B}) = 1, \text{ if } yz \leq \gamma, \text{ where } z = \frac{w^T \bar{x} + b}{\sqrt{w^T \Sigma w}}$$
$$0 \text{ otherwise}$$

Expected error ($\mathbf{e_{exp}}$): It is the ratio of the volume of the ellipsoid on the wrong side of the classifier to the entire volume of the ellipsoid.

Experimental results are reported for a public domain dataset Pima downloaded from UCI repository ([8]). It has 768 observations, where each is observation is a 8-dimensional vector. The dataset had missing entries (marked as 0). To tackle this, we performed the same imputation strategy as in [22]. Then we randomly partitioned the data into test set and training set in the ratio 1 : 9 respectively. We assumed $\Sigma_i = \Sigma$ and $\gamma_i = \gamma$ for all i and we have chosen C via cross-validation. We implemented SVM-OGD and SVM-SOCP both for soft-margin SVM formulation.



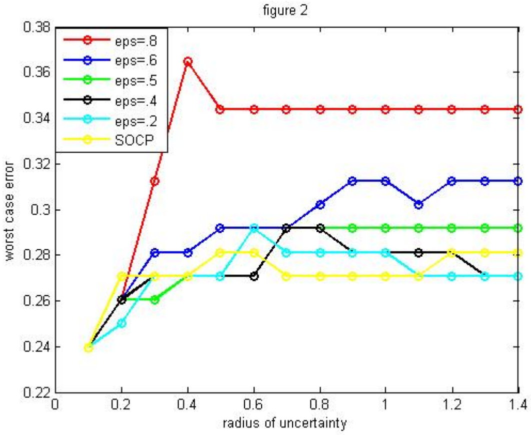
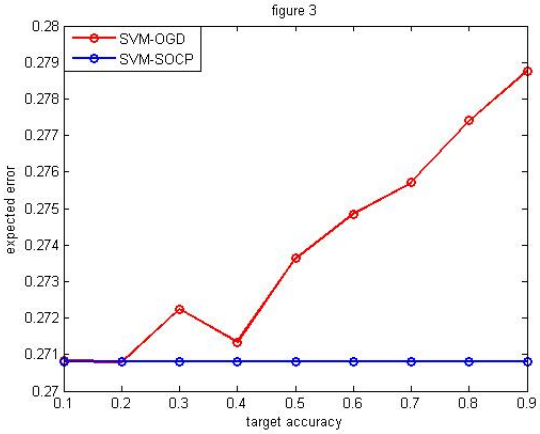


Figure 1 and 2 compares performance of SVM-OGD to SVM-SOCP for different values of target accuracy ε. Figure 1 summarizes how **expected error** varies with the radius (γ) of the uncertainty ellipsoid and figure 2 does the same for **worst case error**. We observe that both the errors generally increases (or remains same) with increasing γ, albeit a few fluctuations. Also as expected for higher values of ε, error is high and for lower values error is low. However, for lower values of ε error of SVM-OGD becomes comparable to SVM-SOCP as we increase γ.



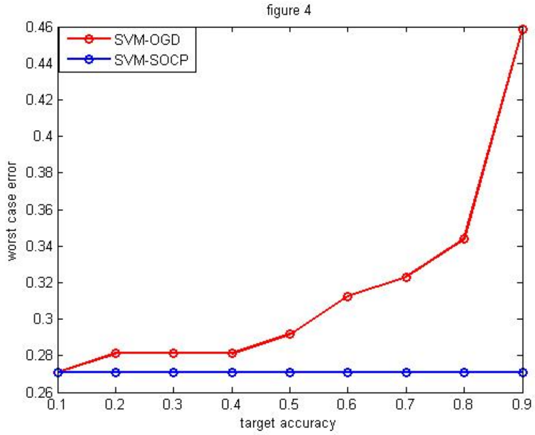


Figure 3 and 4 summarizes how expected error and worst case error of SVM-OGD varies with varying target accuracy ε taking SVM-SOCP as a benchmark. As expected we observe both the error increases as ε increases, while **worst case error** increases much faster than **expected error**.

Our main observation is that both $\mathbf{e_{exp}}$ and $\mathbf{e_{wc}}$ is more for SVM-OGD than SVM-SOCP as the former solves the problem approximately.

8 Conclusion

In this work we considered using online learning approaches for effectively solving robust optimization problems without transforming the problem to a different, more complex, class of problems. We showed that if the problem is a convex-concave saddle-point problem, then we can solve it approximately by playing a repeated game between two players, each choosing his actions following a belief based learning rule, namely fictitious play. Our results suggest $O(1/\sqrt{T})$ rate of convergence, that is after T iterations our approximate solution takes the value of the objective function $O(1/\sqrt{T})$ closer to the optimal value.

Though there exists a better algorithm ([18]) with $O(1/T)$ rate of convergence for solving saddle-point problems, our result is significant in a great sense.

Our algorithm makes use of the fact that in a two player zero-sum game (equivalently saddle-point problem) both players are generally unaware of the structure of the payoff matrix (equivalently the objective function f), or else they will play their Nash equilibrium strategy (optimal strategy) at the first round itself. It operates with the values (and subgradients) of f only, that is each player only gets to see his current payoff based his opponent's previous action, which is closer to reality than knowing the entire game matrix itself even before the start of the game. However in [18], they assumed that we are given the structure of the objective f and thus know the payoff structure beforehand.

More specifically, they assumed that both the players know $\mathscr{W}$ and $\mathscr{U}$ completely and able to compute the value and gradient of f at any point (w,u). But our method assumes that there is a black-box, which given a point (w,u) gives us the value and gradient of f at that point only, which is significant as in zero-sum games the players need not know the complete strategy set of his opponent beforehand. Also, they require f to be $C^{1,1}$, i.e. has Lipschitz continuous gradients, where our method is suitable for any convex-concave function f.

We have proved the convergence of fictitious play for two player zero-sum infinite games, which we believe is a significant extension to the literature itself and there is a scope of further study for any general multiplayer game. Also here we considered simultaneous fictitious play only. It will be interesting to see whether similar results can be obtained for the version of the game where players update their beliefs alternatively. Finally, there is a lot scope of study in identifying problems from diverse backgrounds where our approach can be used efficiently.

References

[1] Aharon Ben-Tal, Sahely Bhadra, Chiranjib Bhattacharyya, and Arkadi Nemirovski. Efficient methods for robust classification under uncertainty in kernel matrices. *The Journal of Machine Learning Research*, 13(1):2923–2954, 2012.

[2] Aharon Ben-Tal, Elad Hazan, Tomer Koren, and Shie Mannor. Oracle-based robust optimization via online learning. *arXiv preprint arXiv:1402.6361*, 2014.

[3] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[4] Ulrich Berger. Brown's original fictitious play. *Journal of Economic Theory*, 135(1):572–578, 2007.

[5] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[6] Sahely Bhadra, Sourangshu Bhattacharya, Chiranjib Bhattacharyya, and Aharon Ben-Tal. Robust formulations for handling uncertainty in kernel matrices. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 71–78, 2010.

[7] Chiranjib Bhattacharyya, KS Pannagadatta, and Alexander J Smola. A second order cone programming formulation for classifying missing data. In *Neural Information Processing Systems (NIPS)*, pages 153–160, 2005.

[8] Catherine Blake and Christopher J Merz. {UCI} repository of machine learning databases. 1998.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] George W Brown. Some notes on computation of games solutions. Technical report, DTIC Document, 1949.

[11] Giuseppe Calafiore and Marco C Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[12] Constantine Caramanis, Shie Mannor, and Huan Xu. 14 robust optimization in machine learning. *Optimization for machine learning*, page 369, 2012.

[13] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[14] Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[15] Samuel Karlin. *Mathematical methods and theory in games, programming, and economics*, volume 2. Courier Corporation, 2003.

[16] Gert Lanckriet, Laurent E Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. Minimax probability machine. In *Advances in Neural Information Processing Systems*, pages 801–807, 2001.

[17] Harry Markowitz. Portfolio selection*. *The journal of finance*, 7(1):77–91, 1952.

[18] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[19] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[20] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[21] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[22] Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, 7:1283–1314, 2006.

[23] TB Trafalis and RC Gilbert. Robust support vector machines for classification and computational issues. *Optimisation Methods and Software*, 22(1):187–198, 2007.

[24] Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10:1485–1510, 2009.

[25] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.