Fast Projection Onto Convex Smooth Constraints

Ilnura Usmanova [1] **Maryam Kamgarpour** [2] **Andreas Krause** [3] **Kfir Yehuda Levy** [4 5]

Abstract

The Euclidean projection onto a convex set is an important problem that arises in numerous constrained optimization tasks. Unfortunately, in many cases, computing projections is computationally demanding. In this work, we focus on projection problems where the constraints are smooth and the number of constraints is significantly smaller than the dimension. The runtime of existing approaches to solving such problems is either cubic in the dimension or polynomial in the inverse of the target accuracy. Conversely, we propose a simple and efficient primal-dual approach, with a runtime that scales only linearly with the dimension, and only logarithmically in the inverse of the target accuracy. We empirically demonstrate its performance, and compare it with standard baselines.

1. INTRODUCTION

Constrained optimization problems arise naturally in numerous fields such as control theory, communication, signal processing, and machine learning (ML). A common approach for solving constrained problems is to *project onto the set of constraints* in each step of the optimization method. Indeed, in ML the most popular learning method is *projected* stochastic gradient descent (SGD). Moreover, projections are employed within projected quasi-Newton (Schmidt et al., 2009), and projected Newton-type methods.

The projection operation in itself requires solving a quadratic optimization problem over the original constraints. In this work, we address the case where we have several smooth constraints, i.e., our constraint set $\mathcal{K}$ is

$$\mathcal{K} = \{x \in \mathbb{R}^n : h_i(x) \le 0\,; \forall i \in [m]\}\,, \quad (1)$$

where h_i's are convex and smooth. We focus on the case where the dimension of the problem n is *high*, and the number of constraints m is *low*. This captures several important ML applications, like multiple kernel learning (Ye et al., 2007), semi-supervised learning (Zhu et al., 2006), triangulation in computer vision (Aholt et al., 2012), applications in signal processing (Huang and Palomar, 2014), solving constrained MDPs (Altman and Asingleutility, 1999; Jin and Sidford, 2020).

In some special cases like box constraints, ℓ_2 or ℓ_1 constraints, the projection problem can be solved very efficiently. Nevertheless, in general there does not exist a unified and scalable approach for projection. One generic family of approaches for solving convex constrained problems are Interior Point Methods (IPM) (Karmarkar, 1984; Nemirovski and Todd, 2008). Unfortunately, in general the runtime of IPMs scales as $O\left(n^3 m \log(n/\varepsilon)\right)$, where ε is the accuracy of the solution, so these methods are unsuitable for high dimensional problems.

Our contribution. We propose a generic and scalable approach for projecting onto a small number of convex smooth constraints. Our approach applies generally for any constraint set that can be described by Eq. (1). Moreover, our approach extends beyond the projection objective to any strongly convex and smooth objective. The overall runtime of our method for finding an approximate projection is $O(nm^{2.5}\log^2(1/\varepsilon) + m^{3.5}\log(1/\varepsilon))$ (see Thm. 3.2 and the discussion afterwards). Thus, the runtime of our method scales *linearly* with n, making it highly suitable for solving high-dimensional problems that are ubiquitous in ML. Furthermore, in contrast to the Frank-Wolfe (FW) algorithm (Frank and Wolfe, 1956), our approach is generic (i.e., does not require a linear minimization oracle) and depends only *logarithmically* on the accuracy.

Moreover, we extend our technique beyond the case of intersections of few smooth constraints. In particular, we provide a conversion scheme that enables to efficiently project onto norm balls using an oracle that projects onto their dual. One can interpret this result as an algorithmic equivalence between projections onto norm ball and its

[1] Automatic Control Laboratory, D-ITET, ETH Zürich, Switzerland [2] Department of Electrical and Computer Engineering, University of British Columbia, Canada [3] Department of Computer Science, ETH Zürich, Switzerland [4] A Viterby fellow [5] Department of Electrical & Computer Engineering, Technion - Israel Institute of Technology. Correspondence to: Ilnura Usmanova <ilnurau@control.ee.ethz.ch>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

dual. This holds for both smooth and non-smooth norms.

On the technical side, our approach utilizes the dual formulation of the problem, and solves it using a cutting plane method. Our key observation is that in the special case of projections, one can efficiently compute approximate gradients and values for the dual problem, which we then use within the cutting plane method. Along the way, we prove the convergence of cutting plane methods with approximate gradient and value oracles, which may be of independent interest.

Related work. In the past years, projection free first order methods have been extensively investigated. In particular, the *Frank-Wolfe* (FW) algorithm (Frank and Wolfe, 1956) (also known as conditional gradient method) is explored, e.g., in (Jaggi, 2013; Garber and Hazan, 2015; Lacoste-Julien and Jaggi, 2015; Garber and Meshi, 2016; Garber, 2016; Lan and Zhou, 2016; Allen-Zhu et al., 2017; Lan et al., 2017). This approach avoids projections and instead assumes that one can efficiently solve an optimization problem with *linear* objective over the constraints in each round. Unfortunately, the latter assumption holds only in special cases. In general, this linear minimization might be non-trivial and have the same complexity as the initial problem. Moreover, FW is in general unable to enjoy the fast convergence rates that apply to standard gradient based methods [1]. In particular, FW does not achieve the linear rate obtained by *projected gradient descent* in the case of smooth and strongly-convex problems. Moreover, FW does not enjoy the accelerated rate obtained by *projected Nesterov's method* for smooth and convex problems.

Further popular approaches for solving constrained problems are the Augmented Lagrangian method and ADMM (Alternating Direction Method of Multipliers) (Boyd et al., 2011; He and Yuan, 2012; Goldstein et al., 2014; Eckstein and Yao, 2012). Such methods work directly on the Lagrangian formulation of the problem while adding penalty terms. Under specific conditions, their convergence rate may be linear (Nishihara et al., 2015; Giselsson and Boyd, 2014). However, ADMM requires the ability to efficiently compute the proximal operator, which as a special case includes the projection operator. To project onto the intersection of convex constraint sets $\cap_{i=1}^{m}\mathcal{K}_i$, consensus ADMM can exploit projection oracles for each $\mathcal{K}_i$ separately. For this general case, only sublinear rate is shown (Xu et al., 2017; Peters and Herrmann, 2019). In the special case of polyhedral sets $\mathcal{K}_i$, it can have linear rate (Hong and Luo, 2017).

Levy and Krause (2019) suggest a fast projection scheme that can approximately solve a projection onto a *single* smooth constraint. However, their approach cannot ensure an arbitrarily small accuracy. Li et al. (2020) extend this approach to a simple constraint like ℓ_1, ℓ_∞-ball in addition to a single smooth constraint. Basu et al. (2017) address high-dimensional QCQPs (Quadratically Constrained Quadratic Programs) via a polyhedral approximation of the feasible set obtained by sampling low-discrepancy sequences. Nevertheless, they only show that their method converges asymptotically, and do not provide any convergence rates. There are also works focusing on fast projections on the sets with a good structure like ℓ_1, ℓ_∞ balls (Condat, 2016; Gustavo et al., 2018; Li and Li, 2020).

Primal-dual formulation of optimization problems is a standard tool that has been extensively explored in the literature. For example, Arora et al. (2005), Plotkin et al. (1995) and Lee et al. (2015) propose to apply the primal-dual approach to solving LPs and SDP. Nevertheless, almost all of the previous works consider problems which are either LPs or SDPs, these are very different from the projection problem that we consider here. In particular:
(i) These works make use of the specialized structure of LP's and SDP's, which does not apply to our work where we consider general constraints. Plotkin et al. (1995) consider general convex constraints, but assume the availability of an oracle that can efficiently solve LP's over this set. This is a very strong assumption that we do not make.

(ii) We devise and employ an approximate gradient oracle for our dual problem is novel way, which is done by a natural combination of Nesterov's method in the primal together with a cutting plane method in the dual. Furthermore, we provide a novel analysis for the projection problem, showing that an approximate solution to the dual problem can be translated to an approximate primal solution.

Thus, the techniques and challenges in our paper are very different from the ones in the aforementioned papers.

Preliminaries and Notation. We denote the Euclidean norm by $\|\cdot\|$. For a positive integer t we denote $[t] = \{1,\ldots,t\}$. A function $F : \mathbb{R}^n \mapsto \mathbb{R}$ is α-*strongly convex* if, $F(y) \geq F(x) + \nabla F(x)^\top(y-x) + \frac{\alpha}{2}\|x-y\|^2$, $\forall x, y \in \mathbb{R}^n$. It is well known that strong-convexity implies $\forall x \in \mathbb{R}^n$, $\frac{\alpha}{2}\|x - x^*\|^2 \leq F(x) - F(x^*)$, where $x^* = \arg\min_{x\in\mathbb{R}^n} F(x)$.

2. PROBLEM FORMULATION

The general problem of Euclidean projection is defined as a constrained optimization problem

$$\min_{x\in\mathbb{R}^n} \quad \|x_0 - x\|^2$$
$$\text{subject to } h_i(x) \leq 0, \ \ \forall i = 1,\ldots,m, \tag{P1}$$

[1] Note that for some special cases like simplex constraints one can ensure fast rates for FW (Garber and Hazan, 2013; Lacoste-Julien and Jaggi, 2015). In general, FW requires $O(1/\varepsilon)$ calls to an oracle providing the solution to linear-minimization oracle to obtain ε-accurate solutions.

where the constraints $h_i : \mathbb{R}^n \to \mathbb{R}$ are convex.

Goal: Our goal in this work is to find an ε-approximate projection $\bar{x}$, such that for any $x : h_i(x) \le 0$ we have, $\|\bar{x} - x_0\|^2 \le \|x - x_0\|^2 + \varepsilon$, and $h_i(\bar{x}) \le \varepsilon , \forall i \in [m]$.

Assumptions: Defining $\mathcal{K} := \{x \in \mathbb{R}^n : h_i(x) \le 0;\ \forall i \in [m]\}$, we assume that $\mathcal{K}$ is compact. Furthermore, we assume the h_i's to be L-smooth and G-Lipschitz continuous in the convex hull of $\mathcal{K}$ and x_0, i.e., $|h_i(x) - h_i(y)| \le G\|x - y\|$, $\forall i \in [m]\ \forall x, y \in Conv\{\mathcal{K}, x_0\}$ and $\|\nabla h_i(x) - \nabla h_i(y)\| \le L\|x - y\|$, $\forall i \in [m]\ \forall x, y \in \mathbb{R}^n$. We assume both G and L to be known. We denote by $H > 0$ the bound $\max_{x \in \mathcal{K}} |h_i(x)| \le H,\ \forall i \in [m]$. We further assume that the distance between x_0 and $\mathcal{K}$ is bounded by B, $\min_{x \in \mathcal{K}} \|x - x_0\| \le B$. Our method does not require the knowledge of B, H. The Lipschitz continuity and smoothness assumptions above are standard and often hold in machine learning applications.

KKT conditions: Our final assumption is that Slater's condition holds, i.e., that there exists a point $x \in \mathbb{R}^n$ such that $\forall i \in [m];\ h_i(x) < 0$. Along with convexity this immediately implies that the optimal solution x^* to Problem (P1) satisfies the KKT conditions, i.e., there exist $\lambda_*^{(1)}, \ldots \lambda_*^{(m)} \in \mathbb{R}_+$ s.t. $(x^* - x_0) + \sum_{i=1}^m \lambda_*^{(i)} \nabla h_i(x^*) = 0$, $\lambda_*^{(i)} h_i(x^*) = 0,\ \forall i \in [m]$, and that there exists a finite bound on $|\lambda_*^{(i)}|\ \forall i \in [m]$. Throughout this paper, we assume the knowledge of an upper bound that we denote by R: $|\lambda_*^{(i)}| \le R\ ; \quad \forall i \in [m]$. In appendix B.1, we showcase two problems where we obtain such a bound explicitly. When such a bound is unknown in advance, one can apply a generic technique to estimating R "on the fly", by applying a standard doubling trick. This will only yield a constant factor increase in the overall runtime. We elaborate on this in appendix B.2. For simplicity we assume throughout the paper that $R \ge 1$.

3. FAST PROJECTION APPROACH

3.1. Intuition: the Case of a Single Constraint

As a warm-up, consider the case of a *single* smooth constraint

$$\min_{x \in \mathbb{R}^n : h(x) \le 0} \quad \|x_0 - x\|^2 . \tag{2}$$

Our fast projection method relies on the (equivalent) dual formulation of the above problem. Let us first define the Lagrangian $\forall x \in \mathbb{R}^n, \lambda \ge 0, \mathcal{L}(x, \lambda) := \|x_0 - x\|^2 + \lambda h(x)$. Note that $\mathcal{L}(\cdot, \cdot)$ is strongly convex in x and concave in λ. Denoting the dual objective by $d(\lambda)$, the dual problem is,

$$\max_{\lambda \ge 0} d(\lambda), \text{ where } d(\lambda) := \min_{x \in \mathbb{R}^n} \|x_0 - x\|^2 + \lambda h(x). \tag{3}$$

We denote an optimal dual solution by $\lambda_* \in \arg\max_{\lambda \ge 0} d(\lambda)$. Our approach is to find an approximate optimal solution to the dual problem $\max_{\lambda \ge 0} d(\lambda)$. Here we show how to do so, and demonstrate how this translates to an approximate solution for the original projection problem (Eq. (2)).

The intuition behind our method is the following. $\mathcal{L}(x, \lambda)$ is linear in λ, and $d(\lambda) := \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda)$, therefore $\max_{\lambda \ge 0} d(\lambda)$ is a *one-dimensional* concave problem. Moreover, $d(\lambda)$ is differentiable and smooth since the primal problem is strongly convex (see Lemma 3.2). Thus, if we could access an *exact* gradient oracle for $d(\cdot)$, we could use *bisection* (see Alg. 6 in the appendix) in order to find an ε-approximate solution to the dual problem within $O(\log(1/\varepsilon))$ iterations (Juditsky, 2015). Due to strong duality, this translates to an ε-approximate solution of the original problem (Eq. (2)). While an exact gradient oracle for $d(\cdot)$ is unavailable, we can efficiently compute *approximate* gradients for $d(\cdot)$. Fixing $\lambda \ge 0$, this can be done by (approximately) solving the following program,

$$\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda) := \|x - x_0\|^2 + \lambda h(x). \tag{4}$$

Letting $x_\lambda^* = \arg\min_{x \in \mathbb{R}^n} \|x - x_0\|^2 + \lambda h(x)$ one can show that $\nabla d(\lambda) = h(x_\lambda^*)$. Thus, in order to devise an approximate estimate for $\nabla d(\lambda)$, it is sufficient to solve the above *unconstrained* program in x to within a sufficient accuracy. This can be done at a linear rate using Nesterov's Accelerated Gradient Descent (AGD) (see Alg. 3) due to the fact that Eq. (4) is a smooth and strongly-convex problem (recall that $h(\cdot)$ is smooth). These approximate gradients can then be used instead of the exact gradients of $d(\cdot)$ to find an ε-optimal solution to the dual problem within $O(\log(1/\varepsilon))$ iterations. The formal description for the case of a single constraint can be found in Appendix A. Next we discuss our approach for the case with several constraints.

Remark Note that using Nesterov's AGD method for the dual problem in the same way as we do for the primal problem sounds like a very natural idea. However, we cannot guarantee the strong concavity of the dual problem and hence, we cannot hope for the linear convergence rate of this approach. In contrast, the bisection algorithm can guarantee the linear convergence rate even for non-strongly concave dual problems.

3.2. Duality of Projections Onto Norm Balls

As a first application, we show how the approach from Section 3.1 has applications for efficient projection on norm balls. The dual of a norm is an important notion that is often used in the analysis of algorithms. Here we show an algorithmic connection between the projection onto norms and onto their dual. Concretely, we show that one can use our framework in order to obtain *an efficient conversion scheme* that enables to project onto a given unit norm ball using an oracle that enables to project onto its dual norm ball. This applies even if the norms are non-smooth, thus extending our technique beyond constraint sets that can be expressed as an intersection of few smooth constraints. Our approach can also be generalized to general convex sets and their dual (polar) sets.

Given a norm $P : \mathbb{R}^n \mapsto \mathbb{R}$, its dual norm is defined as,

$$P_*(x) := \max_{P(z)\le 1} z^\top x \; ; \; \forall x \in \mathbb{R}^n$$

As an example, for any $p \ge 1$ the dual of the ℓ_p-norm is the ℓ_q-norm with $q = p/(p-1)$. Furthermore, the dual of the spectral norm (over matrices) is the nuclear norm; finally for a PD matrix $A \in \mathbb{R}^{d\times d}$ we can define the induced norm $\|x\|_A = x^\top A x$, whose dual is $(\|x\|_A)_* := \|x\|_{A^{-1}} := x^\top A^{-1} x$.

Our goal is to project onto the norm ball w.r.t. $P(\cdot)$, i.e.,

$$\min_{x\in\mathbb{R}^n : P(x)\le 1} \quad \|x_0 - x\|^2 \; . \tag{5}$$

Next we state our main theorem for this section,

Theorem 3.1. *Let $P(\cdot)$ be a norm, and assume that we have an oracle that enables to project onto its dual norm ball $P_*(\cdot)$. Then we can find an ε-approximate solution to Problem (5), by using $O(\log(1/\varepsilon))$ calls to that oracle.*

The idea behind this conversion scheme between norm ball projections is to start with the dual formulation of the problem as we describe in Eq. (3). Interestingly, one can show that the projection oracle onto the dual norm, enables to compute the *exact* gradients of $d(\lambda)$ in this case. This in turn enables to find an approximate solution to the dual problem using only logarithmically many calls to the dual projection oracle. Then we can show that such a solution can be translated to an approximate primal solution. We elaborate on our approach in Appendix C.

3.3. Projecting onto the Intersection of Several Non-Linear Smooth Constraints.

In the rest of this section we will show how to extend our method from Section 3.1 to problems with several constraints. Similarly to Section 3.1, we solve the dual objective using approximate gradients, which we obtain by running Nesterov's method over the primal variable x. Differently from the one-dimensional case, the dual problem is now *multi-dimensional*, so we cannot use bisection. Instead, we employ *cutting plane methods* like center of gravity (Levin, 1965; Newman, 1965), the Ellipsoid method (Shor, 1977; Iudin and Nemirovskii, 1977), and Vaidya's method (Vaidya, 1989). These methods are especially attractive in our context, since their convergence rate depends only *logarithmically* on the accuracy, and their runtime is *linear* in the dimension n. Our main result, Theorem 3.2, states that we find an ε-approximate solution to the projection problem (P1) within a total runtime of $O\left(nm^{3.5}\log(m/\varepsilon) + m^4\log(m/\varepsilon)\right)$ if we use the classical Ellipsoid method, and a runtime of $O\left(nm^{2.5}\log(m/\varepsilon) + m^{3.5}\log(m/\varepsilon)\right)$ if we use the more sophisticated method by Vaidya (1989).

The Lagrangian of the original problem (P1) is defined as follows: $\forall x \in \mathbb{R}^n, \lambda^{(1)}, \ldots, \lambda^{(m)} \ge 0$, $\mathcal{L}(x,\lambda) := \|x - x_0\|^2 + \lambda^\top \mathbf{h}(x)$, where $\lambda := (\lambda^{(1)}, \ldots, \lambda^{(m)}), \mathbf{h}(x) := (h_1(x), \ldots, h_m(x)) \in \mathbb{R}^m$. Defining $d(\lambda) := \min_{x\in\mathbb{R}^n} \mathcal{L}(x,\lambda)$, the dual problem is now defined as follows,

$$\max_{\lambda\in\mathbb{R}^m, \lambda\ge 0} d(\lambda) \; , \tag{6}$$

where $\lambda \ge 0$ is an elementwise inequality. Recall that we assume that we are given $R \ge 0$ such that $\lambda_* \in \{\lambda : \|\lambda\|_\infty \le R\}$, for some $\lambda_* \in \arg\max_{\lambda\ge 0} d(\lambda)$. Thus, our dual problem can be written as

$$\max_{\lambda\in\mathcal{D}} d(\lambda) \; , \tag{P2}$$

where $\mathcal{D} := \{\lambda \in \mathbb{R}^m : \; \forall i \in [m]; \; \lambda^{(i)} \in [0, R]\}$, and $\lambda^{(i)}$ is the i^{th} component of λ. Thus, $\mathcal{D}$ is an ℓ_∞-ball of diameter R centered at $[R/2, \ldots, R/2]^T$. In the rest of this section, we describe and analyze the two components of our fast projection algorithm. In Sec. 3.4 we describe the first component, which is a cutting plane method that we use to solve the dual objective. In contrast to the standard cutting plane approach where *exact* gradient and value oracles are available, we describe and analyze a setting with *approximate oracles*. Next, in Sec. 3.5 we show how to construct approximate gradient and value oracles using a fast first order method. Finally, in Sec. 3.6 we show how to combine these components to our fast projection algorithm that approximately solves the projection problem.

3.4. Cutting Plane Scheme with Approximate Oracles

We first describe a general recipe for cutting plane methods, which captures the center of gravity, Ellipsoid method, and Vaidya's method amongst others. Such schemes require access to exact gradient and value oracles for the objective. Unfortunately, in our setting we are only able to devise *approximate* oracles. To address this issue, we provide a generic analysis, showing that every cutting plane method converges *even when supplied with approximate oracles*. This result may be of independent interest, since it will enable the use of Cutting plane schemes in ML application where we often only have access to approximate oracles.

Cutting Plane Scheme: We seek to solve $\max_{\lambda\in\mathcal{D}} d(\lambda)$, where $\mathcal{D}$ is a compact convex set in $\mathbb{R}^m$ and $d(\cdot)$ is concave. Now, assume that we may access an exact separation oracle $\mathcal{O}_s$ for $\mathcal{D}$, that is, for any $\lambda_t \notin \mathcal{D}$, $\mathcal{O}_s$ outputs $w \in \mathbb{R}^m$ such that $\mathcal{D} \subseteq \{\lambda \in \mathbb{R}^m : w^\top(\lambda - \lambda_t) \leq 0\}$. We also assume access to $(\varepsilon_g, \varepsilon_v)$-approximate gradient and value oracles $\mathcal{O}_g : \mathcal{D} \mapsto \mathbb{R}^m, \mathcal{O}_v : \mathcal{D} \mapsto \mathbb{R}$ for $d(\cdot)$, meaning, $\|\nabla d(\lambda) - \mathcal{O}_g(\lambda)\| \leq \varepsilon_g,\ \ |d(\lambda) - \mathcal{O}_v(\lambda)| \leq \varepsilon_v$. Finally, assume that we are given a point $\lambda_1 = [R/2, \ldots, R/2] \in \mathcal{D}$, and $R > 0$ such that $\mathcal{D} \subseteq M_1 := \{\lambda \in \mathbb{R}_+^m : \|\lambda - \lambda_1\|_\infty \leq R/2\}$. A cutting plane method works as demonstrated in Alg. 1.

Algorithm 1 Cutting Plane Method with Approximate Oracles

Input: gradient and value oracles $\mathcal{O}_g, \mathcal{O}_v$ with accuracies $(\varepsilon_g, \varepsilon_v)$, and exact separation oracle $\mathcal{O}_s$
for $t \in [T]$ **do**
  if $\lambda_t \in \mathcal{D}$ **then**
    call gradient oracle $g_t \leftarrow \mathcal{O}_g(\lambda_t)$, set $w_t = -g_t$;
  else
    call separation oracle and set $w_t \leftarrow \mathcal{O}_s(\lambda_t)$.
  end if
  Construct M_{t+1} such that $\{\lambda \in M_t : w_t^\top(\lambda - \lambda_t) \leq 0\} \subseteq M_{t+1}$, and choose $\lambda_{t+1} \in M_{t+1}$.
end for
Output: $\bar{\lambda} \in \arg\max_{\lambda \in \{\lambda_1, \ldots, \lambda_T\} \cap \mathcal{D}} \mathcal{O}_v(\lambda)$.

Remark 1: The output of the scheme in Alg. 1 is always non-empty since we have assumed $\lambda_1 \in \mathcal{D}$.

Cutting plane methods differ from each other by the construction of sets M_t's and choices of query points λ_t's. For such methods, the volume of M_t's decreases *exponentially* fast with t, and this gives rise to linear convergence guarantees in the case of exact gradient and value oracles.

Definition 3.1 (θ-rate Cutting Plane method)**.** *We say that a cutting plane method has rate $\theta > 0$ if the following applies: $\forall t$, $Vol(M_t)/Vol(M_1) \leq e^{-\theta t}$; and Vol is the usual m-dimensional volume.*

For example, for the *center of mass method* as well as *Vaidya's method*, we have $\theta = O(1)$, for the *Ellipsoid method* we have $\theta = O(1/m)$. Our next lemma extends the convergence of cutting plane methods to the case of approximate oracles. Let us first denote $\mathcal{D}_\varepsilon := \{\lambda \in \mathcal{D} : d(\lambda) \geq d(\lambda_*) - \varepsilon\}$ the set of all ε-approximate solutions , where $\lambda_* \in \arg\max_{\lambda \in \mathcal{D}} d(\lambda)$. We need $\mathcal{D}_\varepsilon$ to have nonzero volume to ensure the required accuracy after the sufficient decrease of volume of M_t. Later we show that in our case with Lipschitz continuous and convex $h_1, \ldots, h_m$, then $\mathcal{D}_\varepsilon$ contains ℓ_∞-ball of radius $r(\varepsilon) \propto \varepsilon/m$ (Corollary 3.1).

Lemma 3.1. *Let $\lambda_1 \in \mathcal{D}, R > 0$ such, $\mathcal{D} \subseteq \{\lambda : \|\lambda - \lambda_1\|_\infty \leq R/2\}$. Given $\varepsilon > 0$ assume that there exists an ℓ_∞-ball of diameter $r(\varepsilon) > 0$ that is contained in $\mathcal{D}_\varepsilon$. Now assume that $d(\lambda)$ is concave and we use the cutting plane scheme of Alg. 1 with oracles that satisfy $\varepsilon_g \leq \frac{\varepsilon}{R\sqrt{m}}$, and $\varepsilon_v \leq \varepsilon$. Then after $T = O(\frac{m}{\theta}\log(R/r(\varepsilon)))$ rounds it outputs $\bar{\lambda} \in \mathcal{D}$ such that, $\max_{\lambda \in \mathcal{D}} d(\lambda) - d(\bar{\lambda}) \leq 4\varepsilon$, where θ is the rate of the cutting plane method.*

Proof. We denote $\mathcal{T}_{\text{Active}} = \{t \in [T] : \lambda_t \in \mathcal{D}\}$, clearly this set is non-empty since $\lambda_1 \in \mathcal{D}$. Also, for any $t \in \mathcal{T}_{\text{Active}}$ we denote $g_t := \mathcal{O}_g(\lambda_t)$ (note that in this case $w_t = -g_t$). We divide the proof into two cases: when $\mathcal{D}_\varepsilon$ is separated by w_t from all $\lambda_t \in \mathcal{D}$, and when not.

Case 1: Assume that there exists $t \in \mathcal{T}_{\text{Active}}$, and $\lambda_\varepsilon \in \mathcal{D}_\varepsilon$ such that, $w_t^\top(\lambda_\varepsilon - \lambda_t) = g_t^\top(\lambda_t - \lambda_\varepsilon) \geq 0$. In this case, using the concavity of $d(\cdot)$ and definitions of g_t, R, we get, $d(\lambda_t) \geq d(\lambda_\varepsilon) + \nabla d(\lambda_t)^\top(\lambda_t - \lambda_\varepsilon) = d(\lambda_\varepsilon) + g_t^\top(\lambda_t - \lambda_\varepsilon) + (\nabla d(\lambda_t) - g_t)^\top(\lambda_t - \lambda_\varepsilon) \geq d(\lambda_*) - \varepsilon - R\sqrt{m}(\varepsilon/(R\sqrt{m})) = d(\lambda_*) - 2\varepsilon$, where we used $\|y\|_2 \leq \sqrt{m}\|y\|_\infty,\ \forall y \in \mathbb{R}^m$. Thus, $d(\bar{\lambda}) \geq \mathcal{O}_v(\bar{\lambda}) - \varepsilon \geq \mathcal{O}_v(\lambda_t) - \varepsilon \geq d(\lambda_t) - 2\varepsilon \geq d(\lambda_*) - 4\varepsilon$.

Case 2: Assume that for any $t \in \mathcal{T}_{\text{Active}}$, and any $\lambda_\varepsilon \in \mathcal{D}_\varepsilon$, we have $w_t^\top(\lambda_\varepsilon - \lambda_t) = g_t^\top(\lambda_t - \lambda_\varepsilon) \leq 0$. This implies that $\forall t \in [T], \forall \lambda_\varepsilon \in \mathcal{D}_\varepsilon$, $w_t^\top(\lambda_\varepsilon - \lambda_t) \leq 0$. Hence $\forall t \in [T],\ \mathcal{D}_\varepsilon \subseteq M_t$, implying that

$$\forall t \in [T]\ \text{Vol}(\mathcal{D}_\varepsilon) \leq\ \text{Vol}(M_t). \tag{7}$$

Next, we show that the above condition can hold only if $T \leq \frac{m}{\theta}\log(R/r(\varepsilon))$. Indeed, according to our assumption $\text{Vol}(\mathcal{D}_\varepsilon) \geq \text{Vol}(\ell_\infty\text{-ball of radius } r(\varepsilon)) = r^m(\varepsilon)$. On the other hand, we assume that $\text{Vol}(M_t) \leq e^{-\theta t}\text{Vol}(\ell_\infty\text{-ball of radius } R/2) = e^{-\theta t}(R/2)^m$. Combining these with Eq. (7) implies that in order to satisfy **Case 2**, we must have $T \leq \frac{m}{\theta}\log(R/2r(\varepsilon))$. Thus, for any $T > \frac{m}{\theta}\log(R/r(\varepsilon))$, **Case 1** must hold, which establishes the lemma. □

3.5. Gradient and Value Oracles for the Dual

Here we show how to efficiently devise gradient and value oracles for the dual objective. Our scheme is described in Alg. 2. Similarly to the one-dimensional case, given λ we approximately minimize $L(\cdot, \lambda)$, which enables us to derive approximate gradient and value oracles. The guarantees of Alg. 2 are given in Lemma 3.3. Before we state the guarantees of Alg. 2 we derive a closed form formula for $\nabla d(\lambda)$. Recall that, $d(\lambda) = \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda)$, and that $\mathcal{L}(x, \lambda)$ is 2-strongly-convex in x. This implies that the minimizer of $\mathcal{L}(\cdot, \lambda)$ is unique, and we therefore denote,

$$x_\lambda^* := \arg\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda)\ . \tag{8}$$

The next lemma shows we can compute $\nabla d(\lambda)$ based on x_λ^*, and states the smoothness of $\nabla d(\lambda)$.

Algorithm 2 $\mathcal{O}$- approximate gradient/value oracles for $d(\cdot)$

Input: $\lambda \geq 0$, target accuracy $\tilde{\varepsilon}$
Compute x_λ, an $\tilde{\varepsilon}$-optimal solution of $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda) := \|x - x_0\|^2 + \lambda^\top \mathbf{h}(x)$.
Method: Nesterov's AGD (Alg. 3) with $\alpha = 2, \beta = 2 + \|\lambda\|_1 L$, and $T = O(\sqrt{\beta} \log(\beta B/\tilde{\varepsilon}))$.
Let: $v := \|x_\lambda - x_0\|^2 + \lambda^\top \mathbf{h}(x_\lambda)$, $g := \mathbf{h}(x_\lambda)$
Output: (x_λ, g, v)

Algorithm 3 Accelerated Gradient Descent (AGD) (Nesterov, 1998)

Input: $F : \mathbb{R}^n \to \mathbb{R}$, $x_0 \in \mathbb{R}^n$, iterations T, strong-convexity α, smoothness β
Set: $y_0 = x_0$, $\kappa := \beta/\alpha$
for $t = 0, \dots, T-1$ **do**
  $y_{t+1} = x_t - \frac{1}{\beta} \nabla F(x_t)$,
  $x_{t+1} = \left(1 + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right) y_{t+1} - \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} y_t$.
end for
Output: y_T

Lemma 3.2. *For any $\lambda \geq 0$ the following holds:*
(i) $\nabla d(\lambda) = \mathbf{h}(x^*_\lambda)$*, and* $\forall \lambda_1, \lambda_2 \geq 0$,

$$\|\nabla d(\lambda_1) - \nabla d(\lambda_2)\| \leq m G^2 \|\lambda_1 - \lambda_2\| ;$$
$$\text{and,} \quad \|x^*_{\lambda_1} - x^*_{\lambda_2}\| \leq \sqrt{m} G \|\lambda_1 - \lambda_2\|.$$

Also, ***(ii)*** $d(\lambda_*) - d(\lambda) \leq m^2 G^2 \|\lambda - \lambda_*\|^2_\infty + mH\|\lambda - \lambda_*\|_\infty$*. Moreover,* ***(iii)*** $x^* = x^*_{\lambda_*}$*, where* λ_*, x^* *are the optimal solutions to the dual and primal problems.*

The proof is quite technical and can be found in Appendix D.1. From the above lemma we can show that for any ε there exists an ℓ_∞-ball of a sufficiently large radius $r(\varepsilon)$ contained in the set of ε-optimal solutions to the dual problem in $\mathcal{D}$.

Corollary 3.1. *Let $\varepsilon \in [0, 1]$. Then there exists an ℓ_∞-ball of radius $r(\varepsilon) := (2m)^{-1} \min\{\varepsilon/H, \sqrt{\varepsilon}/G\}$ that is contained in the set of ε-optimal solutions within $\mathcal{D}$.*

The proof is in Appendix D.2. Eq. (8) together with Lemma 3.2 suggest that exactly minimizing $\mathcal{L}(\cdot, \lambda)$ enables to obtain gradient and value oracles for $d(\cdot)$. In Alg 2 we do so approximately, and the next lemma shows that this translates to approximate oracles.

Lemma 3.3. *Given, $\lambda \geq 0$, running Alg. 2 it outputs, (x, g, v) such that the following applies:*

$$\textit{(i)}\ \|g - \nabla d(\lambda)\| \leq \sqrt{mG^2 \tilde{\varepsilon}}\ ;\ \textit{(ii)}\ \|x - x^*_\lambda\|^2 \leq \tilde{\varepsilon}\ ;$$
$$\text{and } \textit{(iii)}\ |v - d(\lambda)| \leq \tilde{\varepsilon}\ .$$

Additionally, Alg. 2 requires $T_{\text{Internal}} = O(\sqrt{1 + mRL} \log(m/\tilde{\varepsilon}))$ *queries for the gradient of* $\mathbf{h}(\cdot)$*, and its total runtime is* $O(nmT_{\text{Internal}}) \approx O(nm^{3/2} \log(m/\tilde{\varepsilon}))$.

The proof is in Appendix D.3. The proof of the first part is based on 2-strong-convexity of $\mathcal{L}(\cdot, \lambda)$ and G-Lipschitz continuity of $\mathbf{h}(\cdot)$. The second part of the above result also uses the convergence rate of Nesterov's AGD (Nesterov, 1998) described in Appendix in Theorem A.1. Using the notation that appears in the description of the cutting plane method (Alg. 1) we can think of Alg. 2 as a procedure that receives $\lambda \geq 0$ and returns a gradient oracle $\mathcal{O}_g(\lambda) := g$, value oracle $\mathcal{O}_v(\lambda) := v$, and primal solution oracle $\mathcal{O}_x(\lambda) := x_\lambda$.

Remark: Notice that scaling the constraints $\mathbf{h}$ by a factor $\alpha > 0$ leaves the constraints set unchanged, while scaling the smoothness L by a factor of α. Nonetheless, this naturally also scales the bound of the Lagrange multipliers, R, by a factor of $1/\alpha$. Lemma 3.3 tells us that the runtime of our algorithm, T_{Internal}, depends only on RL and is therefore *invariant* to such scaling.

3.6. Fast Projection Algorithm

Below we describe how to compose the two components presented in Sections 3.4 and 3.5 to a complete algorithm for solving the projection problem of (P1).

Algorithm 4 Fast Projection Method

Input: Accuracy parameters $\tilde{\varepsilon} > 0$, $\lambda_1 \in D$, number of rounds T
(1) For any $\lambda \in \mathbb{R}^m$ define three oracles: $\mathcal{O}_g(\lambda) := g$, $\mathcal{O}_v(\lambda) := v$, and $\mathcal{O}_x(\lambda) := x_\lambda$ according to the output (g, v, x_λ) of Alg. 2 with the inputs λ and $\tilde{\varepsilon}$,
(2) Define the separation oracle $\mathcal{O}_s(\lambda)^i := [1, \text{if } \lambda^{(i)} > R; 0, \text{if } \lambda^{(i)} \in (0, R); -1, \text{if } \lambda^{(i)} < 0]$,
(3) Employ a cutting plane method as in Alg. 1 for solving the dual problem, $\max_{\lambda \in \mathcal{D}} d(\lambda)$,
Output: $\bar{\lambda} \in \arg\max_{\lambda \in \{\lambda_1, \dots, \lambda_T\} \cap \mathcal{D}} \mathcal{O}_v(\lambda_t)$, and $\bar{x} = \mathcal{O}_x(\bar{\lambda})$.

Our method in Alg. 4 employs a cutting plane scheme (Alg. 1), while using Alg. 2 in order to devise the gradient and value oracles for $d(\cdot)$. Next we discuss the role of the primal solution oracle $\mathcal{O}_x$, and connect it to our overall projection scheme. Recall that the cutting plane method that we use above finds $\bar{\lambda}$, which is an approximate solution to the dual problem. To extract a primal solution from the dual solution $\bar{\lambda}$, it makes sense to approximately solve $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \bar{\lambda})$, and this is exactly what the oracle $\mathcal{O}_x$ provides (see Alg. 2). Next we state the guarantees of the above scheme.

Theorem 3.2. *Let $\varepsilon > 0$, and consider the projection problem of (P1), and its dual formulation in Eq. (6). Then upon invoking the scheme in Alg. 4 with $\tilde{\varepsilon} = \frac{\varepsilon^4}{256(mRG)^6}$, and $T = O(\frac{m}{\theta} \log(mR/\varepsilon))$, it outputs $\bar{x}$ such that $\forall x \in \mathcal{K} :=$*

$\{x : \mathbf{h}(x) \le 0\}$,

$$\|\bar{x}-x_0\|^2 \le \|x-x_0\|^2+6\varepsilon; \quad \textit{and} \quad h_i(\bar{x}) \le \varepsilon, \ \forall i \in [m] \,.$$

Moreover, the total runtime of our method is $O\left(nm^{2.5}\theta^{-1}\log^2(m/\varepsilon) + \tau_{\text{CP}}(m)m\theta^{-1}\log(mR/\varepsilon)\right)$, *where* θ *is the rate of the cutting plane method (Def. 3.1), and* $\tau_{\text{CP}}(m)$ *is the extra runtime required by the cutting plane method for updating the sets* M_t *beyond calling the gradient and value oracles.*

Let us discuss two choices of a cutting plane method:
Ellipsoid method: In this case $\theta = O(1/m)$ and $\tau_{\text{CP}}(m) = O(m^2)$. Thus, when used within our scheme the total runtime is $O\left(nm^{3.5}\log(m/\varepsilon) + m^4\log(m/\varepsilon)\right)$.
Vaidya's method: In this case $\theta = O(1)$ and $\tau_{\text{CP}}(m) = O(m^{2.5})$. Thus, when used within our scheme the total runtime is $O\left(nm^{2.5}\log(m/\varepsilon) + m^{3.5}\log(m/\varepsilon)\right)$.

Proof of Thm. 3.2. First notice that we may apply the cutting plane method of Alg. 1 since $\mathcal{D}$ is an ℓ_∞-ball of diameter R, so we can set $M_1 := \mathcal{D}$, and λ_1 as its center. Moreover, according to Corollary 3.1 for any $\varepsilon \ge 0$ there exists $r \propto \varepsilon/m$ such that an ℓ_∞-ball of radius r is contained in the set of ε-optimal solutions to the dual problem in $\mathcal{D}$. Let us denote $\bar{\varepsilon} := \left(\frac{\varepsilon}{4mRG}\right)^2$, and notice that we can write $\tilde{\varepsilon} = \left(\frac{\bar{\varepsilon}}{mRG}\right)^2$. Now by setting $\tilde{\varepsilon}$ as accuracy parameter to Alg. 2, it follows from Lemma 3.3 that it generates gradient and value oracles with the following accuracies, $\varepsilon_g = \sqrt{mG^2\tilde{\varepsilon}} \le \frac{\bar{\varepsilon}}{R\sqrt{m}}$; and $\varepsilon_v \le \tilde{\varepsilon} \le \bar{\varepsilon}$. Now applying Lemma 3.1 with these accuracies implies that within $T = \frac{m}{\theta}\log(mR/\varepsilon)$ calls to these approximate oracles it outputs a solution $\bar{\lambda}$ such that $d(\bar{\lambda}) \ge d(\lambda_*) - 4\bar{\varepsilon}$. Next we show that this guarantee on the dual translates to a guarantee for $\bar{x}$ w.r.t. the original primal problem (P1). We will require the following lemma, proved in Appendix D.4.

Lemma 3.4. *Let* $F : \mathbb{R}^m \to \mathbb{R}$ *be an* L*-smooth and concave function, and let* $\lambda_* = \arg\max_{\lambda\in\mathcal{D}} F(\lambda)$*. Also let* $\mathcal{D}$ *is a convex subset of* $\mathbb{R}^m$*. Then,* $\|\nabla F(\lambda) - \nabla F(\lambda_*)\|^2 \le 2L\left(F(\lambda_*) - F(\lambda)\right), \quad \forall \lambda \in \mathcal{D}$.

Using the above lemma together with the mG^2-smoothness of $d(\cdot)$ (Lemma 3.2) implies,

$$\|\nabla d(\bar{\lambda}) - \nabla d(\lambda_*)\| \le \sqrt{8mG^2\bar{\varepsilon}} \,. \tag{9}$$

Now, using $\bar{g} := \mathbf{h}(\bar{x})$ (Alg. 2), and $\|\bar{g} - \nabla d(\bar{\lambda})\| \le \sqrt{mG^2\tilde{\varepsilon}}$ (Lemma 3.3), as well as $\nabla d(\lambda_*) = \mathbf{h}(x^*)$ (Lemma 3.2), we conclude from Eq. (9):

$$\begin{aligned}&\|\mathbf{h}(\bar{x}) - \mathbf{h}(x^*)\| \le \|\mathbf{h}(\bar{x}) - \nabla d(\bar{\lambda})\| + \|\nabla d(\bar{\lambda}) - \mathbf{h}(x^*)\| \\ &= \|\bar{g} - \nabla d(\bar{\lambda})\| + \|\nabla d(\bar{\lambda}) - \nabla d(\lambda_*)\| \le \sqrt{16mG^2\bar{\varepsilon}} \,,\end{aligned} \tag{10}$$

where we used $\tilde{\varepsilon} \le \bar{\varepsilon}$. The above implies that $\forall i \in [m]$, $h_i(\bar{x}) = h_i(x^*) + (h_i(\bar{x}) - h_i(x^*)) \le h_i(x^*) + |h_i(\bar{x}) - h_i(x^*)| \le 0 + \|\mathbf{h}(\bar{x}) - \mathbf{h}(x^*)\|_\infty \le mG\sqrt{16\bar{\varepsilon}} \le \varepsilon$, where the second inequality uses the feasibility of x^*, and the last line uses the definition of $\bar{\varepsilon}$ (we assume $R \ge 1$). This concludes the first part of the proof. Moreover, from Eq. (10) we also get,

$$\begin{aligned}&-\bar{\lambda}^\top\mathbf{h}(\bar{x}) \\ &= -\bar{\lambda}^\top(\mathbf{h}(\bar{x}) - \mathbf{h}(x^*)) - (\bar{\lambda} - \lambda_*)^\top\mathbf{h}(x^*) - (\lambda_*)^\top\mathbf{h}(x^*) \\ &\le \sqrt{16mG^2\bar{\varepsilon}}\|\bar{\lambda}\| + \nabla d(\lambda_*)^\top(\lambda_* - \bar{\lambda}) + 0 \\ &\le mG\sqrt{16\bar{\varepsilon}}\|\bar{\lambda}\|_\infty + d(\lambda_*) - d(\bar{\lambda}) \\ &\le mGR\sqrt{16\bar{\varepsilon}} + 4\bar{\varepsilon} \le 5\varepsilon \,.\end{aligned} \tag{11}$$

where the first inequality uses Eq. (10) as well as $\mathbf{h}(x^*) = \nabla d(\lambda_*)$ (Lemma 3.2) and complementary slackness, which implies $(\lambda_*)^\top\mathbf{h}(x^*) = 0$; the second inequality uses the concavity of $d(\cdot)$ implying that $d(\lambda_*) - d(\bar{\lambda}) \ge \nabla d(\lambda_*)^\top(\lambda_* - \bar{\lambda})$, and the last line uses the definition of $\bar{\varepsilon}$ as well as $\bar{\varepsilon} \le \varepsilon$. Using Eq. (11) together with $\tilde{\varepsilon}$-optimality of $\bar{x}$ with respect to $\mathcal{L}(\cdot, \bar{\lambda})$ (Alg. 2) implies that $\forall x \in \mathcal{K} := \{x : h_i(x) \le 0; \ \forall i \in [m]\}$ we have, $\|\bar{x} - x_0\|^2 \le \|x - x_0\|^2 + \bar{\lambda}^\top\mathbf{h}(x) - \bar{\lambda}^\top\mathbf{h}(\bar{x}) + \tilde{\varepsilon} \le \|x - x_0\|^2 + 6\varepsilon$, and we used $\tilde{\varepsilon} \le \varepsilon$, and $\bar{\lambda} \ge 0, \mathbf{h}(x) \le 0$. This concludes the proof.
Runtime: for a single $t \in [T]$ we invoke Alg. 2, and its runtime is $O(nm^{1.5}\log(m/\varepsilon))$ (Lemma 3.3), additionally τ_{CP} for the update. Multiplying this by T we get a runtime of $O(nm^{2.5}\theta^{-1}\log^2(m/\varepsilon) + \tau_{CP}m\theta^{-1}\log(m/\varepsilon))$. Also, every call to the separation oracle for $\mathcal{D}$ takes $O(m)$ which is negligible compared to computing the gradient and value oracles. □

Note that inside our algorithm we could use not only Vaidya's and Ellipsoid methods, but any other cutting plane scheme. For example, the faster cutting plane methods proposed by Lee et al. (2015), Jiang et al. (2020) can be used as well.

4. EXPERIMENTAL EVALUATION

4.1. Synthetic Problem

We first demonstrate the performance of our approach on synthetic problems of projection onto a randomly generated quadratic set and onto their intersection.

$$\begin{aligned}&\min_{x\in\mathbb{R}^n} \|x - x_p\|^2 \\ &\text{subject to } (x - x_i)^T A_i(x - x_i) \le 0, \ i = 1, \dots, m.\end{aligned} \tag{12}$$

The matrices A_i are generated randomly in such a way that they are positive definite and have norm equal to 1. We compare our approach with the Interior Point Method

(IPM) from the MOSEK solver, as well as with SLSQP from the *scipy.optimize.minimize* package. For Algorithm 2, to solve the primal subproblems we use the AGD method as described before. We select the smoothness parameter L is based on the norms of the matrices A_i, and tune the parameter R empirically using the doubling trick.The run-times are shown in Table 4.1. The run-times are averaged over 5 runs of the method on the random inputs. The accuracy is fixed to 10^{-4}. The results demonstrate a substantial performance improvement obtained by our fast projection approach as the dimensionality increases. The runtime in seconds is not a perfect performance measure, but is the most reasonable measure we could think of. Comparing the number of iterates hides the complexity of each iteration which might be huge for interior point methods.

4.2. Learning the Kernel Matrix in Discriminant Analysis via QCQP (Kim et al., 2006; Ye et al., 2007; Basu et al., 2017)

We next consider an application in multiple kernel learning. Consider a standard binary classification setup where $\mathcal{X}$ – a subset of $\mathbb{R}^n$ – denotes the input space, and $\mathcal{Y} = \{-1, +1\}$ denotes the output (class label) space. We assume that the examples are independently drawn from a fixed unknown probability distribution over $\mathcal{X} \times \mathcal{Y}$. We model our data with positive definite *kernel* functions (Schölkopf et al., 2018). In particular, for any $x_1, \dots, x_n \in \mathcal{X}$, the *Gram* matrix, defined by $G_{jk} = K(x_j, x_k)$ is positive semi-definite. Let $X = [x_1^+, \dots, x_{n_+}^+, x_1^-, \dots, x_{n_-}^-]$ be a data matrix of size $n = n_+ + n_-$, where $\{x_1^+, \dots, x_{n_+}^+\}$ and $\{x_1^-, \dots, x_{n_-}^-\}$ are the data points from positive and negative classes. For binary classification, the problem of kernel learning for discriminant analysis seeks, given a set of p kernel matrices $G^i = K^i(x_j, x_k), x_j, x_k \in X, i \in [p], G^i \in \mathbb{R}^{n\times n}$ to learn an optimal linear combination $G \in \mathcal{G} = \left\{G \,|\, G = \sum_{i=1}^p \theta_i G^i, \sum_{i=1}^p \theta_i = 1, \theta_i \geq 0\right\}$. This problem was introduced by Fung et al. (2004), reformulated as an SDP by Kim et al. (2006), and as a much more tractable QCQP by Ye et al. (2007). Latter approach learns an optimal kernel matrix $\tilde{G} \in \tilde{\mathcal{G}} = \left\{\tilde{G} \,|\, \tilde{G} = \sum_{i=1}^p \theta_i \tilde{G}^i, \sum_{i=1}^p \theta_i r_i = 1, \theta_i \geq 0\right\}$, where $\tilde{G}^i = G^i P G^i, r_i = \text{Trace}(\tilde{G}^i)$, $P = I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$, and $\mathbf{1}_n$ is the vector of all ones of size n, by solving the following convex QCQP

$$\max_{\beta,t} \; -\frac{1}{4}\beta^T\beta + \beta^T a - \frac{\lambda}{4}t \tag{13}$$

$$\text{subject to } \; t \geq \frac{1}{r_i}\beta^T\tilde{G}^i\beta, \; i = 1, \dots, p, \tag{14}$$

where $a = [1/n^+, \dots, 1/n^+, -1/n^-, \dots, -1/n^-] \in \mathbb{R}^n, \beta \in \mathbb{R}^n$. Hereby λ is a regularization parameter that we set to $\lambda = 10^{-4}$. The optimal θ corresponds to the dual solution of the above problem (13). Note that in this application, the number of data points n is much larger than the number of constraints (i.e., the number of kernel matrices), making it ideally suited for our approach. We run our algorithm applied for this problem over β with fixed $t = 5\cdot 10^{-8}$. Then the problem becomes strongly convex: $\arg\max_\beta -\frac{1}{4}\beta^T\beta + \beta^T a + \lambda^T t = \arg\max_\beta -\frac{1}{4}(\beta^T\beta - 4\beta^T a + 4a^T a) = \arg\max_\beta -\frac{1}{4}\|\beta - 2a\|_2^2$. We use the *doc-rna* dataset (Uzilov et al., 2006) from LIBSVM with $n = 4000, 10000, 11000$ data points and compare the results and the running time with the IPM. We focus on learning a convex combination of m Gaussian Kernels $K(x,z) = \sum_{i=1}^m \theta_i e^{-\|x-z\|^2/\sigma_i^2}$ with different bandwidth parameters σ_i, chosen uniformly on the logarithmic scale over the interval $[10^{-1}, 10^2]$, as in (Kim et al., 2006; Ye et al., 2007). Results are shown in Table 4.2 below. Moreover, for the Kernel Learning problem with $\tilde{\varepsilon} = 500\varepsilon^2$, we present the results for IPM and Fast Projection algorithms for $m = 3, n = 11000$ dependent on the target accuracy in Table 3. Note that quadratic constraints do not satisfy Lipschitz continuity assumption on the whole $\mathbb{R}^n$. However, the Lipschitz continuity holds on any compact set inside $\mathbb{R}^n$. Since the AGD algorithm keeps the iterates on the compact set, this is enough to guarantee the Lipschitz continuity. Moreover, the Lipschitz constant G itself is needed only to specify the accuracy for AGD $\tilde{\varepsilon}$. It only affects the runtime of AGD logarithmically. The parameter H is not needed to be known since it influences only the upper bound on the runtime of the ellipsoid method. [2]

5. CONCLUSION

We proposed a novel method for fast projection onto smooth convex constraints. We employ a primal-dual approach, and combine cutting plane schemes with Nesterov's accelerated gradient descent. We analyze its performance and prove its effectiveness in high-dimensional settings with a small number of constraints. The results are generalizable to any strongly-convex objective with smooth convex constraints. Our work demonstrates applicability of cutting plane algorithms in the field of Machine Learning and can potentially improve efficiency of solving high dimensional constrained optimization problems. Enforcing constraints can be of crucial importance when ensuring reliability and safety of machine learning systems.

Acknowledgements

We thank the reviewers for the helpful comments. This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No

[2]The experiments were run on a machine with Intel Core i7-7700K, 64Gb RAM.

Table 1. Run-times (in seconds). Hereby, n is the dimensionality of the problem, the number of constraints is 2.

m = 2, n:	10	100	500	1000	2000	5000	8000	10000	12000
SLSQP	0.011	0.384	3.481	9.757	47.143	573.980	-	-	-
IPM	0.059	0.073	0.577	2.427	11.850	118.414	408.137	751.216	901.878
Fast Proj	0.416	2.429	3.746	15.504	22.482	141.704	350.681	547.240	666.231
ADMM	23.761	92.836	285.383	-	-	-	-	-	-

Table 2. Run-times (in seconds). Hereby, n is the number of data points (dimensionality), the number of kernels is 3 (number of constraints). For large problems, our approach outperforms IPM.

m = 3, n:	4000	10000	11000
Fast Proj	230.281	768.086	1216.9440
IPM	75.133	906.631	1302.088

Table 3. Run-times (in seconds). Hereby, ε is the target accuracy in objective value, the number of kernels is 3 (number of constraints). For large problems and smaller accuracies, our approach outperforms IPM.

ε	10^{-6}	10^{-7}	10^{-8}
Fast Proj	83.2381	519.7166	1216.9440
IPM	1011.5410	1070.3363	1302.0882

815943, the Swiss National Science Foundation under the grant SNSF 200021_172781, and under NCCR Automation under grant agreement 51NF40 180545, as well as the Israel Science Foundation (grant No. 447/20).

References

Aholt, C., Agarwal, S., and Thomas, R. (2012). A qcqp approach to triangulation. In *European Conference on Computer Vision*, pages 654–667. Springer.

Allen-Zhu, Z., Hazan, E., Hu, W., and Li, Y. (2017). Linear convergence of a frank-wolfe type algorithm over trace-norm balls. In *Advances in Neural Information Processing Systems*, pages 6192–6201.

Altman, E. and Asingleutility, I. (1999). Constrained markov decision processes.

Arora, S., Hazan, E., and Kale, S. (2005). Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 339–348. IEEE.

Basu, K., Saha, A., and Chatterjee, S. (2017). Large-scale quadratically constrained quadratic program via low-discrepancy sequences. In *Advances in Neural Information Processing Systems*, pages 2297–2307.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.

Condat, L. (2016). Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, 158(1):575–585.

Eckstein, J. and Yao, W. (2012). Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32(3).

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Fung, G., Dundar, M., Bi, J., and Rao, B. (2004). A fast iterative algorithm for fisher discriminant using heterogeneous kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 40.

Garber, D. (2016). Faster projection-free convex optimization over the spectrahedron. In *Advances in Neural Information Processing Systems*, pages 874–882.

Garber, D. and Hazan, E. (2013). Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE.

Garber, D. and Hazan, E. (2015). Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, pages 541–549.

Garber, D. and Meshi, O. (2016). Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Advances in Neural Information Processing Systems*, pages 1001–1009.

Giselsson, P. and Boyd, S. P. (2014). Metric selection in douglas-rachford splitting and admm. *arXiv: Optimization and Control*.

Goldstein, T., O'Donoghue, B., Setzer, S., and Baraniuk, R. (2014). Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623.

Gustavo, C., Wohlberg, B., and Rodriguez, P. (2018). Fast projection onto the ℓ_∞, ℓ_1-mixed norm ball using steffensen root search. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4694–4698. IEEE.

He, B. and Yuan, X. (2012). On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709.

Hong, M. and Luo, Z. (2017). On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162:165–199.

Huang, Y. and Palomar, D. P. (2014). Randomized algorithms for optimal solutions of double-sided qcqp with applications in signal processing. *IEEE Transactions on Signal Processing*, 62(5):1093–1108.

Iudin, D. and Nemirovskii, A. (1977). Evaluation of informational complexity of mathematical-programming programs. *Matekon*, 13(2):3–25.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435.

Jiang, H., Lee, Y. T., Song, Z., and Wong, S. C.-w. (2020). An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 944–953.

Jin, Y. and Sidford, A. (2020). Efficiently solving mdps with stochastic mirror descent. In *International Conference on Machine Learning*, pages 4890–4900. PMLR.

Juditsky, A. (2015). Convex optimization ii: Algorithms. `https://ljk.imag.fr/membres/Anatoli.Iouditski/cours/convex/chapitre_22.pdf`.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311.

Kim, S.-J., Magnani, A., and Boyd, S. (2006). Optimal kernel selection in kernel fisher discriminant analysis. In *Proceedings of the 23rd international conference on Machine learning*, pages 465–472.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504.

Lan, G., Pokutta, S., Zhou, Y., and Zink, D. (2017). Conditional accelerated lazy stochastic gradient descent. In *International Conference on Machine Learning*, pages 1965–1974.

Lan, G. and Zhou, Y. (2016). Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409.

Lee, Y. T., Sidford, A., and Wong, S. C.-w. (2015). A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065. IEEE.

Levin, A. Y. (1965). An algorithm for minimizing convex functions. In *Doklady Akademii Nauk*, volume 160, pages 1244–1247. Russian Academy of Sciences.

Levy, K. Y. and Krause, A. (2019). Projection free online learning over smooth sets. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Li, Q. and Li, X. (2020). Fast projection onto the ordered weighted ℓ_1 norm ball. *arXiv preprint arXiv:2002.05004*.

Li, Y., Cao, X., and Chen, H. (2020). Fully projection-free proximal stochastic gradient method with optimal convergence rates. *IEEE Access*, 8:165904–165912.

Mahdavi, M., Yang, T., Jin, R., Zhu, S., and Yi, J. (2012). Stochastic gradient descent with only one projection. In *Advances in Neural Information Processing Systems*, pages 494–502.

Nemirovski, A. S. and Todd, M. J. (2008). Interior-point methods for optimization. *Acta Numerica*, 17:191–234.

Nesterov, Y. (1998). Introductory lectures on convex programming volume i: Basic course.

Newman, D. J. (1965). Location of the maximum on unimodal surfaces. *Journal of the ACM (JACM)*, 12(3):395–398.

Nishihara, R., Lessard, L., Recht, B., Packard, A., and Jordan, M. I. (2015). A general analysis of the convergence of admm. *arXiv preprint arXiv:1502.02009*.

Peters, B. and Herrmann, F. J. (2019). Algorithms and software for projections onto intersections of convex and non-convex sets with applications to inverse problems. *arXiv preprint arXiv:1902.09699*.

Plotkin, S. A., Shmoys, D. B., and Éva Tardos (1995). Fast approximation algorithms for fractional packing and covering problems.

Schmidt, M., Berg, E., Friedlander, M., and Murphy, K. (2009). Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *Artificial Intelligence and Statistics*, pages 456–463.

Schölkopf, B., Smola, A. J., and Bach, F. (2018). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

Shor, N. Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1):94–96.

Uzilov, A. V., Keegan, J. M., and Mathews, D. H. (2006). Detection of non-coding rnas on the basis of predicted secondary structure formation free energy change. *BMC Bioinformatics, 7(173)*.

Vaidya, P. M. (1989). A new algorithm for minimizing convex functions over convex sets. In *30th Annual Symposium on Foundations of Computer Science*, pages 338–343. IEEE.

Xu, Z., Taylor, G., Li, H., Figueiredo, M. A. T., Yuan, X., and Goldstein, T. (2017). Adaptive consensus ADMM for distributed optimization. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3841–3850. PMLR.

Yang, T., Lin, Q., and Zhang, L. (2017). A richer theory of convex constrained optimization with reduced projections and improved rates. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3901–3910. JMLR. org.

Ye, J., Ji, S., and Chen, J. (2007). Learning the kernel matrix in discriminant analysis via quadratically constrained quadratic programming. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 854–863.

Zhu, X., Kandola, J., Lafferty, J., and Ghahramani, Z. (2006). Graph kernels by spectral transforms. *Semi-supervised learning*, pages 277–291.