An Efficient Re-scaled Perceptron Algorithm for Conic Systems

Alexandre Belloni [⋆], Robert M. Freund [⋆⋆], and Santosh S. Vempala [⋆⋆⋆]

Abstract. The classical perceptron algorithm is an elementary algorithm for solving a homogeneous linear inequality system $Ax > 0$, with many important applications in learning theory (e.g., [11, 8]). A natural condition measure associated with this algorithm is the Euclidean width τ of the cone of feasible solutions, and the iteration complexity of the perceptron algorithm is bounded by $1/\tau^2$. Dunagan and Vempala [5] have developed a re-scaled version of the perceptron algorithm with an improved complexity of $O(n \ln(1/\tau))$ iterations (with high probability), which is theoretically efficient in τ, and in particular is polynomial-time in the bit-length model. We explore extensions of the concepts of these perceptron methods to the general homogeneous conic system $Ax \in \textbf{int}\ K$ where K is a regular convex cone. We provide a conic extension of the re-scaled perceptron algorithm based on the notion of a *deep-separation oracle* of a cone, which essentially computes a certificate of strong separation. We give a general condition under which the re-scaled perceptron algorithm is theoretically efficient, i.e., polynomial-time; this includes the cases when K is the cross-product of half-spaces, second-order cones, and the positive semi-definite cone and more generally when we have suitable access to both K and its dual cone K^*.

1 Introduction.

We consider the problem of computing a solution of the following conic system

$$\begin{cases} Ax \in \textbf{int}\ K \\ x \in X \end{cases} \tag{1}$$

where X and Y are n- and m-dimensional Euclidean subspaces, respectively, $A : X \to Y$ is a linear operator and $K \subset Y$ is a regular closed convex cone. We refer to this problem as the "conic inclusion" problem, we call K the *inclusion cone* and we call $\mathcal{F} := \{x \in X : Ax \in K\}$ the *feasibility cone.* The goal is to compute an interior element of the feasibility cone $\mathcal{F}$. Important special cases of this format include feasibility problem instances for linear programming (LP), second-order cone programming (SOCP) and positive semi-definite programming (SDP). These problems are often encountered in learning theory, e.g., to learn

[⋆] MIT and IBM, belloni@mit.edu

[⋆⋆] MIT, rfreund@mit.edu

[⋆⋆⋆] Georgia Tech and MIT, vempala@cc.gatech.edu

threshold functions and in support vector machines, to mention two well-known examples.

The ellipsoid method ([10]), the random walk method ([2]), and interior-point methods (IPMs) ([9], [12]) are examples of methods which solve (1) in polynomial-time. These methods differ substantially in their representation requirement as well as in their practical performance. For example, a membership oracle suffices for the ellipsoid method and the random walk method, while a special barrier function for K is required to implement an IPM. The latter is by far the most successful algorithm for conic programming in practice: for example, applications of SDP range over several fields including optimal control, eigenvalue optimization, combinatorial optimization and many others, see [18].

For the important special case of linear inequalities, when $X = \mathbb{R}^n$ and $K = \mathbb{R}^m_+$, an alternative method is the perceptron algorithm [17, 13], developed primarily in learning theory. It is well-known that this simple method terminates after a finite number of iterations which can be bounded by the square of the inverse of the *width* τ of the feasibility cone $\mathcal{F}$. Although attractive due to its simplicity and its noise-tolerance [4, 3], the perceptron algorithm is not considered theoretically efficient since the width τ can be exponentially small in the size of the instance in the bit-length model. Dunagan and Vempala ([5]) combined the perceptron algorithm with a sequence of re-scalings constructed from near-feasible solutions. These re-scalings gradually increase τ on average and the resulting re-scaled perceptron algorithm has complexity $O(n \ln(1/\tau))$ iterations (with high probability), which is theoretically efficient.

Here we extend the re-scaled perceptron algorithm proposed in [5] to the conic setting of (1). Although the probabilistic analysis is similar, this is not the case for the remainder of the analysis. In particular, we observe that the improvement obtained in [5] arises from a clever use of a *deep-separation oracle* (see Def. 3), which is stronger than the usual separation oracle used in the classical perceptron algorithm. In the case of a system of linear inequalities studied in [5], there is no difference between the implementation of both oracles. However, this difference is quite significant for more general cones.

We investigate, in detail, ways to construct a deep-separation oracle for several classes of cones, since it is the driving force of the re-scaled perceptron algorithm. We establish important properties of the deep-separation oracle and its implementation for several classes. Our main technical result is a general scheme that yields a polynomial-time deep-separation oracle using only a deep-separation oracle for the dual cone of K (which is readily available for many cones of interest such as the cone of positive semi-definite matrices). This implies that the re-scaled perceptron algorithm runs in polynomial time for any conic program, provided we have a deep separation oracle for the dual cone of K. This readily captures the important cases of linear programs, second-order cone programs and semi-definite programs [1] and thus conveys the benefits of the perceptron algorithm to these problems.

[1] There have been earlier attempts to extend the algorithm of [5], to SDPs in particular, but unfortunately these have turned out to be erroneous.

We start in Section 2 with properties of convex cones, oracles, and the definition of a deep-separation oracle. Section 3 generalizes the classical perceptron algorithm to the conic setting, and Section 4 extends the re-scaled perceptron algorithm of [5] to the conic setting. Section 5 contains the probabilistic and complexity analysis of the re-scaled perceptron algorithm, which reviews some material from [5] for completeness. Section 6 is devoted to methods for constructing a deep-separation oracle for both specific and general cones. We conclude the introduction with an informal discussion of the main ideas and technical difficulties encountered in obtaining our results.

The perceptron algorithm is a greedy procedure that updates the current proposed solution by using any violated inequality. The number of iterations is finite but can be exponential. The modified perceptron algorithm (proposed in [3], used in [5]) is a similar updating procedure that only uses inequalities that are violated by at least some fixed threshold. Although this procedure is not guaranteed to find a feasible solution, it finds a near-feasible solution with the guarantee that no constraint is violated by more than the threshold and the number of steps to convergence is proportional to the inverse square of the threshold, independent of the conditioning of the initial system. The key idea in [5] is that such a near-feasible solution can be used to improve the width of the original system by a multiplicative factor. As we show in this paper, this analysis extends naturally to the full generality of conic systems.

The main difficulty is in identifying a constraint that is violated by more than a fixed threshold by the current proposed solution, precisely what we call a deep-separation oracle. This is not an issue in the linear setting (one simply checks each constraint). For conic systems, the deep-separation itself is a conic feasibility problem! It has the form: find $w \in K^*$, the dual of the original inclusion cone, such that w satisfies a single second-order conic constraint. Our idea is to apply the re-scaled percepron algorithm to this system which is considerably simpler than $\mathcal{F}$. What we can prove is that provided K^* has a deep-separation oracle, the method is theoretically efficient. For many interesting inclusion cones, including the cone of positive semi-definite matrices, such a deep-separation oracle is readily available.

2 Preliminaries

Let X and Y denote Euclidean spaces with finite dimension n and m, respectively. Denote by $\|\cdot\|$ their Euclidean norms, and $\langle\cdot,\cdot\rangle$ their Euclidean inner products. For $\bar{x} \in X$, $B(\bar{x}, r)$ will denote the ball centered at $\bar{x}$ with radius r, and analogously for Y. Let $A : X \to Y$ denote a linear operator, and $A^* : Y \to X$ denote the adjoint operator associated with A.

2.1 Convex Cones

Let C be a convex cone. The dual cone of C is defined as

$$C^* = \{d : \langle x, d\rangle \geq 0, \text{ for all } x \in C\} \tag{2}$$

and $\mathbf{ext}C$ denote the set of extreme rays of C. A cone is pointed if it contains no lines. We say that C is a *regular* cone if C is a pointed closed convex cone with non-empty interior. It is elementary to show that C is regular if and only if C^* is regular. Given a regular convex cone C, we use the following geometric (condition) measure:

Definition 1. *If C is a regular cone in X, the* width *of C is given by*

$$\tau_C \triangleq \max_{x,r} \left\{ \frac{r}{\|x\|} : B(x,r) \subset C \right\} .$$

Furthermore the center *of C is any vector $\bar{z}$ that attains the above maximum, normalized so that $\|\bar{z}\| = 1$.*

We will be particularly interested in the following three classes of cones: the non-negative orthant $\mathbb{R}^m_+ := \{x \in \mathbb{R}^m : x \geq 0\}$, the second order cone denoted by $Q^n := \{x \in \mathbb{R}^n : \|(x_1, x_2, \ldots, x_{n-1})\| \leq x_n\}$, and the cone of positive semi-definite matrices $S^{k\times k}_+ := \{X \in S^{k\times k} : \langle v, Xv\rangle \geq 0 \text{ for all } v \in \mathbb{R}^k\}$ where $S^{k\times k} := \{X \in \mathbb{R}^{k\times k} : X = X^T\}$. These three cones are self-dual and their widths are $1/\sqrt{m}$, $1/\sqrt{2}$, and $1/\sqrt{k}$, respectively.

The following characterization will be used in our analysis.

Lemma 1. *Let $\mathcal{G} = \{x : Mx \in C\}$ and Let $T = \{M^*\lambda : \lambda \in C^*\}$. Then $\mathcal{G}^* =$ $\mathbf{cl}\ (T)$.*

Proof. ($\subseteq$) Let $\lambda \in C^*$. Then for every x satisfying $Mx \in C$, $\langle x, A^*\lambda\rangle = \langle Ax, \lambda\rangle \geq 0$, since $Mx \in C$ and $\lambda \in C^*$. Thus, $\mathbf{cl}\ (T) \subseteq \mathcal{G}^*$ since $\mathcal{G}^*$ is closed.

($\supseteq$) Assume that there exists $y \in \mathcal{G}^* \backslash \mathbf{cl}\ (T)$. Thus there exists $h \neq 0$ satisfying $\langle h, y\rangle < 0$ and $\langle h, w\rangle \geq 0$ for all $w \in \mathbf{cl}\ (T)$. Notice that $\langle h, M^*\lambda\rangle \geq 0$ for all $\lambda \in C^*$, which implies that $Mh \in C$ and so $h \in \mathcal{G}$. On the other hand, since $y \in \mathcal{G}^*$, it follows that $\langle h, y\rangle \geq 0$, contradicting $\langle h, y\rangle < 0$.

The question of sets of the form T being closed has been recently studied by Pataki [14]. Necessary and sufficient conditions for T to be a closed set are given in [14] when C^* belongs to a class called "nice cones," a class which includes polyhedra and self-scaled cones. Nonetheless, the set T may fail to be closed even in simple cases.

The following property of convex cones is well-known.

Lemma 2. *$B(z,r) \subseteq C$ if and only if $\langle d, z\rangle \geq r\|d\|$ for all $d \in C^*$.*

2.2 Oracles

In our algorithms and analysis we will distinguish two different types of oracles.

Definition 2. *An* interior separation oracle *for a convex set $S \subset \mathbb{R}^n$ is a subroutine that given a point $x \in \mathbb{R}^n$, identifies if $x \in \mathbf{int}\ S$ or returns a vector $d \in \mathbb{R}^n$, $\|d\| = 1$, such that*

$$\langle d, x\rangle \leq \langle d, y\rangle \ \text{ for all } y \in S\ .$$

Definition 3. *For a fixed positive scalar t, a* deep-separation oracle *for a cone $C \subset \mathbb{R}^n$ is a subroutine that given a non-zero point $x \in \mathbb{R}^n$, either*

(I) *correctly identifies that* $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \geq -t$ *for all* $d \in \mathbf{ext}C^*$ *or*

(II) *returns a vector* $d \in C^*$, $\|d\| = 1$ *satisfying* $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \leq -t$.

Definition 2 is standard in the literature, whereas Definition 3 is new as far as we know. Our motivation for this definition arises from a relaxation of the orthogonality characterization of a convex cone. For $d, x \neq 0$ let $\cos(d, x)$ denote the cosine of the angle between d and x, i.e., $\cos(d, x) = \frac{\langle d,x\rangle}{\|d\|\|x\|}$. Notice that $x \in C$ if and only if $\cos(d, x) \geq 0$ for all $d \in C^*$ if and only if $\cos(d, x) \geq 0$ for all $d \in \mathbf{ext}C^*$. The latter characterization states that $\frac{\langle d,x\rangle}{\|d\|\|x\|} \geq 0$ for all $d \in \mathbf{ext}C^*$. Condition (I) of the deep-separation oracle relaxes the cosine condition from 0 to $-t$. The following example illustrates that the perceptron improvement algorithm described in [5] corresponds to a deep-separation oracle for a linear inequality system.

Example 1. Let $C = \{x \in \mathbb{R}^n : Mx \geq 0\}$ where M is an $m \times n$ matrix none of whose rows are zero. Notice that $C^* = \{M^*\lambda : \lambda \geq 0\}$ is the conic hull of the rows of M, and the extreme rays of C^* are a subset of the rows of M. Therefore a deep-separation oracle for C can be constructed by identifying for a given $x \neq 0$ if there is an index $i \in \{1, \ldots, m\}$ for which $\frac{\langle M_i,x\rangle}{\|M_i\|\|x\|} \leq -t$ and returning $M_i/\|M_i\|$ in such a case. Notice that we do not need to know which vectors M_i are extreme rays of C^*; if m is not excessively large it is sufficient to simply check the aforementioned inequality for every row index i.

3 Perceptron Algorithm for a Conic System

The classical perception algorithm was proposed to solve a homogeneous system of linear inequalities (1) with $K = \mathbb{R}^m_+$. It is well-known that the algorithm has finite termination in at most $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations, see Rosenblatt 1962 [17]. This complexity bound can be exponential in the bit-model.

Our starting point herein is to show that the classical perceptron algorithm can be easily extended to the case of a conic system of the form (1).

```
Perceptron Algorithm for a Conic System
(a) Let $x$ be the origin in $X$. Repeat:
(b) If $Ax \in \mathbf{int}\, K$, Stop. Otherwise, call interior separation oracle for $\mathcal{F}$
    at $x$, returning $d \in \mathcal{F}^*$, $\|d\| = 1$, such that $\langle d, x\rangle \leq 0$, and set $x \leftarrow x + d$.
```

This algorithm presupposes the availability of a separation oracle for the feasibility cone $\mathcal{F}$. In the typical case when the inclusion cone K has an interior separation oracle, this oracle can be used to construct an interior separation oracle for $\mathcal{F}$: if $x \notin \textbf{int}\ \mathcal{F}$, then $Ax \notin \textbf{int}\ K$ and there exists $\lambda \in K^*$ satisfying $\langle \lambda, Ax \rangle \leq 0$, whereby $d = A^*\lambda / \|A^*\lambda\|$ satisfies $\langle d, x \rangle \leq 0$, $d \in \mathcal{F}^*$, and $\|d\| = 1$.

Exactly as in the case of linear inequalities, we have the following guarantee. It's proof is identical, via the potential function $\pi(x) = \langle x, \bar{z} \rangle / \|x\|$.

Lemma 3. *The perceptron algorithm for a conic system will compute a solution of (1) in at most $\lfloor 1/\tau_{\mathcal{F}}^2 \rfloor$ iterations.*

4 Re-scaled Conic Perceptron Algorithm

In this section we construct a version of the perceptron algorithm whose complexity depends only logarithmically on $1/\tau_{\mathcal{F}}$. To accomplish this we will systematically re-scale the system (1) using a linear transformation related to a suitably constructed random vector that approximates the center $\bar{z}$ of $\mathcal{F}$. The linear transformation we use was first proposed in [5] for the case of linear inequality systems (i.e., $K = \mathbb{R}^m_+$). Here we extend these ideas to the conic setting. Table 1 contains a description of our algorithm, which is a structural extension of the algorithm in [5].

Note that the perceptron improvement phase requires a deep-separation oracle for $\mathcal{F}$ instead of the interior separation oracle for $\mathcal{F}$ as required by the perceptron algorithm. For the remainder of this section we presuppose that a deep-separation for $\mathcal{F}$ is indeed available. In Section 6 we will show that for most standard cones K a deep-separation oracle for $\mathcal{F}$ can be efficiently constructed.

We begin the analysis with the following lemma that quantifies the impact of the re-scaling (Step 6) on the width of the feasibility cone $\mathcal{F}$.

Lemma 4. *Let $\bar{z}$ denote the center of the feasibility cone $\mathcal{F}$, normalized so that $\|\bar{z}\| = 1$. Let A, $\hat{A}$ denote the linear operators and $\tau_{\mathcal{F}}$, $\tau_{\hat{\mathcal{F}}}$ denote the widths of the feasibility cones $\mathcal{F}$, $\hat{\mathcal{F}}$ of two consecutive iterations of the re-scaled perception algorithm. Then*

$$\tau_{\hat{\mathcal{F}}} \geq \frac{(1-\sigma)}{\sqrt{1+3\sigma^2}\|\hat{z}\|}\tau_{\mathcal{F}}$$

where $\hat{z} = \bar{z} + \frac{1}{2}\left(\tau_{\mathcal{F}} - \left\langle \frac{x}{\|x\|}, \bar{z} \right\rangle\right)\frac{x}{\|x\|}$, and x is the output of the perceptron improvement phase.

Proof. At the end of the perception improvement phase, we have a vector x satisfying

$$\frac{\langle d, x \rangle}{\|d\|\|x\|} \geq -\sigma \quad \text{for all } d \in \textbf{ext}\mathcal{F}^*.$$

Let $\bar{x} = x/\|x\|$. Then $\langle d, \bar{x} \rangle \geq -\sigma\|d\|$ for all $d \in \textbf{ext}\mathcal{F}^*$. From Lemma 2, it holds that

$$\frac{\langle d, \bar{z} \rangle}{\|d\|\|\bar{z}\|} = \frac{\langle d, \bar{z} \rangle}{\|d\|} \geq \tau_{\mathcal{F}} \quad \text{for all } d \in \mathcal{F}^*,$$

i.e. $\langle d, \bar{z} \rangle \geq \tau_{\mathcal{F}}\|d\|$ for all $d \in \mathcal{F}^*$.

```
Re-scaled Perceptron Algorithm for a Conic System

Step 1 Initialization. Set $B = I$ and $\sigma = 1/(32n)$.

Step 2 Perceptron Algorithm for a Conic System.
(a) Let $x$ be the origin in $X$. Repeat at most $\lfloor (1/\sigma^2) \rfloor$ times:
    (b) If $Ax \in \mathbf{int}\ K$, Stop. Otherwise, call interior separation oracle for $\mathcal{F}$ at $x$,
        returning $d \in \mathcal{F}^*$, $\|d\| = 1$, such that $\langle d, x \rangle \leq 0$, and set $x \leftarrow x + d$.

Step 3 Stopping Criteria. If $Ax \in \mathbf{int}\ K$ then output $Bx$ and Stop.

Step 4 Perceptron Improvement Phase.
(a) Let $x$ be a random unit vector in $X$. Repeat at most $\lfloor (1/\sigma^2)\ln(n) \rfloor$ times:
    (b) Call deep-separation oracle for $\mathcal{F}$ at $x$ with $t = \sigma$.
        If $\langle d, x \rangle \geq -\sigma\|d\|\|x\|$ for all $d \in \mathbf{ext}\mathcal{F}^*$ (condition I), End Step 4.
        Otherwise, oracle returns $d \in \mathcal{F}^*$, $\|d\| = 1$, such that $\langle d, x \rangle \leq -\sigma\|d\|\|x\|$
          (condition II), and set $x \leftarrow x - \langle d, x \rangle d$.
        If $x = 0$ restart at (a).
(c) Call deep-separation oracle for $\mathcal{F}$ at $x$ with $t = \sigma$.
    If oracle returns condition (II), restart at (a).

Step 5 Stopping Criteria. If $Ax \in \mathbf{int}\ K$ then output $Bx$ and Stop.

Step 6 Re-scaling. $A \leftarrow A \circ \left( I + \frac{xx^T}{\langle x, x \rangle} \right)$,   $B \leftarrow B \circ \left( I + \frac{xx^T}{\langle x, x \rangle} \right)$,
    and Goto Step 2.
```

Table 1. One iteration of the re-scaled perceptron algorithm is one pass of **Steps 2-6**.

From Lemma 1 it therefore holds that

$$\langle \lambda, A\bar{z} \rangle = \langle A^*\lambda, \bar{z} \rangle \geq \tau_{\mathcal{F}} \|A^*\lambda\| \quad \text{for all} \quad \lambda \in K^*.$$

Note that $\hat{z} = \bar{z} + \frac{1}{2}(\tau_{\mathcal{F}} - \langle \bar{x}, \bar{z} \rangle)\bar{x}$, and let $\hat{\tau} := \frac{(1-\sigma)}{\sqrt{1+3\sigma^2}}\tau_{\mathcal{F}}$. We want to show that

$$\langle v, \hat{z} \rangle \geq \hat{\tau}\|v\| \quad \text{for all} \quad v \in \mathbf{ext}\mathcal{F}^*. \tag{3}$$

If (3) is true, then by convexity of the function $f(v) = \hat{\tau}\|v\| - \langle v, \hat{z} \rangle$ it will also be true that $\langle v, \hat{z} \rangle \geq \hat{\tau}\|v\|$ for any $v \in \mathcal{F}^*$. Then from Lemma 2 it would follow that $B(\hat{z}, \hat{\tau}) \subset \mathcal{F}$, whereby $\tau_{\hat{\mathcal{F}}} \geq \frac{\hat{\tau}}{\|\hat{z}\|}$ as desired.

Let v be an extreme ray of $\mathcal{F}^*$. Using Lemma 1, there exist a sequence $\{\lambda^i\}_{i \geq 1}$, $\lambda^i \in K^*$, $A^*\lambda^i \to v$ as $i \to \infty$. Since (3) is trivially true for $v = 0$, we can assume that $v \neq 0$ and hence $A^*\lambda^i \neq 0$ for i large enough. Next note that

$$\begin{aligned} \|\hat{A}^*\lambda^i\|^2 &= \|A^*\lambda^i\|^2 + 2\left\langle A^*\lambda^i, \bar{x}\right\rangle^2 + \langle \bar{x}, \bar{x}\rangle \left\langle A^*\lambda^i, \bar{x}\right\rangle^2 \\ &= \|A^*\lambda^i\|^2 \left(1 + 3\left(\frac{\left\langle A^*\lambda^i, \bar{x}\right\rangle}{\|A^*\lambda^i\|}\right)^2\right) \end{aligned}$$

and

$$\begin{aligned} \left\langle \hat{A}^*\lambda^i, \hat{z}\right\rangle &= \left\langle A^*\lambda^i, \hat{z}\right\rangle + \langle \bar{x}, \hat{z}\rangle \left\langle A^*\lambda^i, \bar{x}\right\rangle \\ &= \left\langle A^*\lambda^i, \bar{z}\right\rangle + (\tau_{\mathcal{F}} - \langle \bar{x}, \bar{z}\rangle)\left\langle A^*\lambda^i, \bar{x}\right\rangle + \langle \bar{x}, \bar{z}\rangle \left\langle A^*\lambda^i, \bar{x}\right\rangle \\ &\geq \tau_{\mathcal{F}}\|A^*\lambda^i\| + \tau_{\mathcal{F}}\left\langle A^*\lambda^i, \bar{x}\right\rangle \\ &= \tau_{\mathcal{F}}\left(1 + \frac{\left\langle A^*\lambda^i, \bar{x}\right\rangle}{\|A^*\lambda^i\|}\right)\|A^*\lambda^i\|. \end{aligned} \tag{4}$$

Therefore $\dfrac{\left\langle \hat{A}^*\lambda^i, \hat{z}\right\rangle}{\|\hat{A}^*\lambda^i\|} \geq \tau_{\mathcal{F}}\dfrac{1+t_i}{\sqrt{1+3t_i^2}}$ where $t_i = \frac{\left\langle A^*\lambda^i, \bar{x}\right\rangle}{\|A^*\lambda^i\|}$. Note that $t_i \leq 1$ and $\langle v, \bar{x}\rangle \geq -\sigma\|v\|$ since $v \in \mathbf{ext}\mathcal{F}^*$, and so $\dfrac{\langle v, \bar{x}\rangle}{\|v\|} \geq -\sigma$. By continuity, for any $\varepsilon > 0$ it holds that $t_i \geq -\sigma - \varepsilon$ for i sufficiently large. Thus, $t_i \in [-\sigma - \varepsilon, 1]$ for i large enough.

For $t \in [0, 1]$, we have $\frac{1+t}{\sqrt{1+3t^2}} \geq \frac{1+t}{\sqrt{1+2t+t^2}} = 1$, and for $t \in [-\sigma - \varepsilon, 0]$, the function $g(t) = \frac{1+t}{\sqrt{1+3t^2}} \geq \frac{1-\sigma-\varepsilon}{\sqrt{1+3(\sigma+\varepsilon)^2}}$ since

$$\frac{dg(t)}{dt} = \frac{1-3t}{(1+3t^2)^{3/2}} \geq 0$$

for $t \in [-\sigma - \varepsilon, 0]$, that is, $g(t)$ is increasing on $[-\sigma - \varepsilon, 0]$. Therefore, for i large enough we have

$$\frac{\left\langle \hat{A}\lambda^i, \hat{z}\right\rangle}{\|\hat{A}^*\lambda^i\|} \geq \tau_{\mathcal{F}}\frac{(1-\sigma-\varepsilon)}{\sqrt{1+3(\sigma+\varepsilon)^2}}.$$

Passing to the limit as $\lambda^i \to v$ obtain

$$\frac{\langle v, \hat{z}\rangle}{\|v\|} \geq \tau_{\mathcal{F}}\frac{(1-\sigma-\varepsilon)}{\sqrt{1+3(\sigma+\varepsilon)^2}}$$

whereby

$$\frac{\langle v, \hat{z}\rangle}{\|v\|} \geq \tau_{\mathcal{F}}\frac{(1-\sigma)}{\sqrt{1+3\sigma^2}} = \hat{\tau}.$$

5 Probabilistic Analysis.

As mentioned before, the probabilistic analysis of our conic framework is similar to the analysis with linear inequalities in [5]. We state the main lemmas of the analysis without proof. Our exposition intentionally separates the probabilistic analysis from the remaining sections.

The first lemma of this section was established in [3] for the case of linear inequalities, and here is generalized to the conic framework. Roughly speaking, it shows that the perceptron improvement phase generates near-feasible solutions if started at a good initial point, which happens with at least a fixed probability $p = 1/8$.

Lemma 5. *Let z be a feasible solution of (1) of unit norm. With probability at least $\frac{1}{8}$, the perception improvement phase returns a vector x satisfying:*

(i) $\langle d, x\rangle \geq -\sigma\|x\|$ *for every* $d \in \mathbf{ext}\mathcal{F}^*$, $\|d\| = 1$, *and*
(ii) $\langle z, x/\|x\|\rangle \geq \frac{1}{\sqrt{n}}$.

Lemma 5 establishes that points obtained after the perceptron improvement phase are near-feasible for the current conic system. The next lemma clarifies the implications of using these near-feasible points to re-scale the conic system.

Lemma 6. *Suppose that $n \geq 2$, $\tau_{\mathcal{F}}, \sigma \leq 1/32n$ and A is the linear operator of the current iteration. Let $\hat{A}$ be the linear operator obtained after one iteration of the perceptron improvement phase. Let $\tau_{\hat{\mathcal{F}}}$ denote the width of the cone of feasible solutions $\hat{\mathcal{F}}$ of the updated conic system associated with $\hat{A}$. Then*

(i) $\tau_{\hat{\mathcal{F}}} \geq \left(1 - \dfrac{1}{32n} - \dfrac{1}{512n^2}\right)\tau_{\mathcal{F}}$;
(ii) With probability at least $\frac{1}{8}$, $\tau_{\hat{\mathcal{F}}} \geq \left(1 + \dfrac{1}{3.02n}\right)\tau_{\mathcal{F}}$.

Finally, the following theorem bounds the number of overall iterations and the number of oracle calls made by the algorithm.

Theorem 1. *Suppose that $n \geq 2$. If (1) has a solution, the re-scaled perceptron algorithm will compute a solution in at most*

$$T = \max\left\{4096 \ln\left(\frac{1}{\delta}\right), 139n\ln\left(\frac{1}{32n\tau_{\mathcal{F}}}\right)\right\} = O\left(n\ln\left(\frac{1}{\tau_{\mathcal{F}}}\right) + \ln\left(\frac{1}{\delta}\right)\right)$$

iterations, with probability at least $1-\delta$. Moreover, the algorithm makes at most $O(T\ n^2\ \ln(n))$ calls of a deep-separation oracle for $\mathcal{F}$ and at most $O(T\ n^2)$ calls of a separation oracle for $\mathcal{F}$ with probability at least $1 - \delta$.

It will useful to amend Definition 3 of the deep-separation oracle as follows:

Definition 4. *For a fixed positive scalar σ, a* half-deep-separation oracle *for a cone $C \subset \mathbb{R}^n$ is a subroutine that given a non-zero point $x \in \mathbb{R}^n$, either*

(I) correctly identifies that $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \geq -\sigma$ *for all* $d \in \mathbf{ext}C^*$ *or*

(II) returns a vector $d \in C^*$, $\|d\| = 1$ *satisfying* $\dfrac{\langle d, x\rangle}{\|d\|\|x\|} \leq -\sigma/2$.

Remark 1. Definition 4 only differs from Definition 3 in the inequality in condition (II), where now $\sigma/2$ is used instead of σ. This minor change only affects the iteration bound in Step 4 of the re-scaled perceptron algorithm, which needs to be changed to $\lfloor (4/\sigma^2)\ln(n) \rfloor$; all other analysis in this Section remains valid.

6 Deep-separation Oracles

The re-scaled perceptron algorithm needs a deep-separation oracle for the feasibility cone $\mathcal{F}$. Herein we show that such a deep-separation oracle is fairly easy to construct when (1) has the format:

$$\begin{cases} A_L x \in \textbf{int } \mathrm{I\!R}^m_+ \\ A_i x \in \textbf{int } Q^{n_i} \quad i = 1, \ldots, q \\ x_s \in \textbf{int } S^{k\times k}_+ \ , \end{cases} \tag{5}$$

where x is composed as the cartesian product $x = (x_s, x_p)$. Note that (5) is an instance of (1) for $K = \mathrm{I\!R}^m_+ \times Q^{n_1} \times \cdots \times Q^{n_q} \times S^{k\times k}_+$ and the only special structure on A is that the semi-definite inclusion is of the simple format "$I x_s \in S^{k\times k}_+$." In Section 6.4 we show how to construct a deep-separation oracle for more general problems that also include the semi-definite inclusion "$A_s x \in S^{k\times k}_+$," but this construction takes more work.

The starting point of our analysis is a simple observation about intersections of feasibility cones. Suppose we have available deep-separation oracles for each of the feasibility cones $\mathcal{F}_1$ and $\mathcal{F}_2$ of instances:

$$\begin{cases} A_1 x \in \textbf{int } K_1 \\ x \in X \end{cases} \qquad \text{and} \qquad \begin{cases} A_2 x \in \textbf{int } K_2 \\ x \in X \end{cases} \tag{6}$$

and consider the problem of finding a point that simultaneously satisfies both conic inclusions:

$$\begin{cases} A_1 x \in \textbf{int } K_1 \\ A_2 x \in \textbf{int } K_2 \\ x \in X \ . \end{cases} \tag{7}$$

Let $\mathcal{F} = \{x : A_1 x \in K_1, A_2 x \in K_2\} = \{x : Ax \in K\}$ where $K = K_1 \times K_2$ and A is defined analogously. Then $\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2$ where $\mathcal{F}_i = \{x : A_i x \in K_i\}$ for $i = 1, 2$. It follows from the calculus of convex cones that $\mathcal{F}^* = \mathcal{F}_1^* + \mathcal{F}_2^*$, and therefore

$$\textbf{ext}\mathcal{F}^* \subset (\textbf{ext}\mathcal{F}_1^* \cup \textbf{ext}\mathcal{F}_2^*) \ . \tag{8}$$

This observation leads to an easy construction of a deep-separation oracle for $\mathcal{F}_1 \cap \mathcal{F}_2$ if one has available deep-separation oracles for $\mathcal{F}_1$ and $\mathcal{F}_2$:

```
Deep-separation Oracle for $\mathcal{F}_1 \cap \mathcal{F}_2$
Given: scalar $t > 0$ and $x \neq 0$, call deep-separation oracles for $\mathcal{F}_1$ and $\mathcal{F}_2$ at $x$.
    If both oracles report Condition I, return Condition I.
    Otherwise at least one oracle reports Condition II and provides
    $d \in \mathcal{F}_i^* \subset \mathcal{F}^*$, $\|d\| = 1$, such that $\langle d, x\rangle \leq -t\|d\|\|x\|$; return $d$ and Stop.
```

Remark 2. If deep-separation oracles for $\mathcal{F}_i$ are available and their efficiency is $O(T_i)$ operations for $i = 1, 2$, then the deep-separation oracle for $\mathcal{F}_1 \cap \mathcal{F}_2$ given above is valid and its efficiency is $O(T_1 + T_2)$ operations.

Utilizing Remark 2, in order to construct a deep-separation oracle for the feasibility cone of (5) it will suffice to construct deep-separation oracles for each of the conic inclusions therein, which is what we now examine.

6.1 Deep-separation Oracle for $\mathcal{F}$ when $K = \mathbb{R}^m_+$

We consider $\mathcal{F} = \{x : Ax \in \mathbb{R}^m_+\}$. Example 1 has already described a deep-separation oracle for $\mathcal{F}$ when the inclusion cone is $\mathbb{R}^m_+$. It is easy to see that this oracle can be implemented in $O(mn)$ operations.

6.2 Deep-separation Oracle for $\mathcal{F}$ when $K = Q^k$

For convenience we amend our notation so that $\mathcal{F} = \{x : \|Mx\| \leq g^T x\}$ for a given real $(k-1) \times n$ matrix M and a real n-vector g, so that $\mathcal{F} = \{x : Ax \in Q^k\}$ where the linear operator A is specified by $Ax := \begin{bmatrix} Mx \\ g^T x \end{bmatrix}$.

We will construct an efficient half-deep-separation oracle (Definition 4) by considering the following optimization problem:

$$\begin{array}{rcl} t^* & := & \min_d d^T x \\ & & \text{s.t. } \|d\| = 1 \\ & & \quad d \in \mathcal{F}^* \ . \end{array} \tag{9}$$

If $x \in \mathcal{F}$, then $t^* \geq 0$ and clearly condition I of Definition 4 is satisfied. If $x \notin \mathcal{F}$, then $t^* < 0$ and we can replace the equality constraint in (9) with an inequality constraint. We obtain the following primal/dual pair of convex problems with common optimal objective function value t^*:

$$\begin{array}{rcl} t^* := \min_d x^T d & = & \max_y -\|y - x\| \\ \text{s.t. } \|d\| \leq 1 & & \text{s.t. } y \in \mathcal{F} \\ d \in \mathcal{F}^* & & \end{array} \tag{10}$$

Now consider the following half-deep-separation oracle for $\mathcal{F}$ when $K = Q^k$.

```
Half-Deep-Separation Oracle for $\mathcal{F}$
 when $K = Q^k$, for $x \neq 0$ and relaxation parameter $\sigma > 0$
 If $\|Mx\| \leq g^T x$, return Condition I, and Stop.
 Solve (10) for feasible primal and dual solutions $\bar{d}, \bar{y}$ with duality gap $\bar{g}$
 satisfying $\bar{g}/\|x\| \leq \sigma/2$
   If $x^T \bar{d}/\|x\| \geq -\sigma/2$, report Condition (I), and Stop.
   If $x^T \bar{d}/\|x\| \leq -\sigma/2$, then return $d = \bar{d}$, report Condition (II), and Stop.
```

To see the validity of this method, note that if $\|Mx\| \le g^T x$, then $x \in \mathcal{F}$ and clearly Condition (I) of Definition 4 is satisfied. Next, suppose that $x^T \bar{d}/\|x\| \ge -\sigma/2$, then $t^* \ge -\|\bar{y} - x\| = x^T \bar{d} - \bar{g} \ge -\|x\|\sigma/2 - \|x\|\sigma/2 = -\|x\|\sigma$. Therefore $\frac{x^T d}{\|x\|\|d\|} \ge -\sigma$ for all $d \in \mathcal{F}^*$, and it follows that Condition (I) of Definition 4 is satisfied. Finally, if $x^T \bar{d}/\|x\| \le -\sigma/2$, then $\frac{\bar{d}^T x}{\|\bar{d}\|\|x\|} \le -\sigma/2$ and $\bar{d} \in \mathcal{F}^*$, whereby Condition (II) of Definition 4 is satisfied using $\bar{d}$.

The computational efficiency of this deep-separation oracle depends on the ability to efficiently solve (10) for feasible primal/dual solutions with duality gap $\bar{g} \le \sigma\|x\|/2$. For the case when $K = Q^k$, it is shown in [1] that (10) can be solved very efficiently to this desired duality gap, namely in $O(n^3 + n \ln\ln(1/\sigma) + n \ln\ln(1/\min\{\tau_{\mathcal{F}}, \tau_{\mathcal{F}^*}\}))$ operations in practice, using a combination of Newton's method and binary search. Using $\sigma = 1/(32n)$ this is $O(n^3 + n \ln\ln(1/\min\{\tau_{\mathcal{F}}, \tau_{\mathcal{F}^*}\}))$ operations for the relaxation parameter σ needed by the re-scaled perceptron algorithm.

6.3 Deep-separation Oracle for $S_+^{k\times k}$

Let $C = S_+^{k\times k}$, and for convenience we alter our notation herein so that $X \in S^{k\times k}$ is a point under consideration. A deep-separation oracle for C at $X \ne 0$ for the scalar $t > 0$ is constructed by simply checking the condition "$X + t\|X\|I \succeq 0$." If $X + t\|X\|I \succeq 0$, then condition I of the deep-separation oracle is satisfied. This is true because the extreme rays of C are the collection of rank-1 matrices vv^T, and

$$\frac{\langle vv^T, X\rangle}{\|X\|\|vv^T\|} = \frac{v^T X v}{\|X\|\|vv^T\|} \ge \frac{-t\|X\|v^T v}{\|X\|\|vv^T\|} = -t$$

for any $v \ne 0$. On the other hand, if $X + t\|X\|I \not\succeq 0$, then compute any nonzero v satisfying $v^T X v + t\|X\|v^T v \le 0$, and return $D = vv^T/v^T v$, which will satisfy

$$\frac{\langle D, X\rangle}{\|X\|\|D\|} = \frac{v^T X v}{\|X\|v^T v} \le -t \ ,$$

thus satisfying condition II. Notice that the work per oracle call is simply to check the eigenvalue condition $X \succeq -t\|X\|I$ and possibly to compute an appropriate vector v, which is typically $O(k^3)$ operations in practice.

6.4 Methodology for a Deep-separation Oracle for $\mathcal{F}$ when K^* has a Deep-Separation Oracle

In this subsection we present a general result on how to construct a deep-separation oracle for *any* feasibility cone $\mathcal{F} = \{x \in \mathbb{R}^n : Ax \in K\}$ whose dual inclusion cone K^* has an efficiently-computable deep-separation oracle. We therefore formally define our working premise for this subsection as follows:
Premise: K^* has an efficiently-computable deep-separation oracle. Furthermore, τ_K and τ_{K^*} are known.

Remark 3. The results herein specify to the case when $K = S_+^{k\times k}$. We know from the results in Section 6.3 and the self-duality of $S_+^{k\times k}$ ($(S_+^{k\times k})^* = S_+^{k\times k}$) that K^* has an efficiently computable deep-separation oracle when $K = S_+^{k\times k}$. Furthermore, we have $\tau_K = \tau_{K^*} = 1/\sqrt{k}$.

The complexity analysis that we develop in this subsection uses the data-perturbation condition measure model of Renegar [15], which we now briefly review. Considering (1) as a system with fixed cone K and fixed spaces X and Y, let $\mathcal{M}$ denote those operators $A : X \to Y$ for which (1) has a solution. For $A \in \mathcal{M}$, let $\rho(A)$ denote the "distance to infeasibility" for (1), namely:

$$\rho(A) := \min_{\Delta A} \left\{ \|\Delta A\| : A + \Delta A \notin \mathcal{M} \right\} \ .$$

Then $\rho(A)$ denotes the smallest perturbation of our given operator A which would render the system (1) infeasible. Next let $\mathcal{C}(A)$ denote the *condition measure* of (1), namely $\mathcal{C}(A) = \|A\|/\rho(A)$, which is a scale-invariant reciprocal of the distance to infeasibility. $\ln(\mathcal{C}(A))$ is tied to the complexity of interior-point methods and the ellipsoid method for computing a solution of (1), see [16] and [6].

Given a regular inclusion cone K, the feasibility cone for (1) is $\mathcal{F} = \{x : Ax \in K\}$. Given the relaxation parameter $t > 0$ and a non-zero vector $x \in \mathbb{R}^n$, consider the following conic feasibility system in the variable d:

$$(S_{t,x}) : \quad \begin{cases} \frac{\langle x, d\rangle}{\|x\|\|d\|} < -t \\ \\ \qquad d \in \mathcal{F}^* \end{cases} \tag{11}$$

It follows from Definition 3 that if d is feasible for $(S_{t,x})$, then Condition II of Definition 3 is satisfied; however, if $(S_{t,x})$ has no solution, then Condition I is satisfied. Utilizing Lemma 1 and rearranging terms yields the equivalent system in variables w:

$$(S_{t,x}) : \quad \begin{cases} t\|x\|\|A^*w\| + \langle w, Ax\rangle < 0 \\ \\ \qquad\qquad\qquad w \in \mathrm{int} K^* \end{cases} \tag{12}$$

Note that if $\tilde{w}$ solves (12), then $\tilde{d} = A^*\tilde{w}$ solves (11) from Lemma 1. This leads to the following approach to constructing a deep-separation oracle for $\mathcal{F}$:

> given $x \neq 0$ and $t := \sigma$, compute a solution $\tilde{w}$ of (12) or certify that no solution exists. If (12) has no solution, report Condition I and Stop; otherwise (12) has a solution $\tilde{w}$, return $d := A^*\tilde{w}/\|A^*\tilde{w}\|$, report Condition II, and Stop.

In order to implement this deep-separation oracle we need to be able to compute a solution $\tilde{w}$ of (12) if such a solution exists, or be able to provide a certificate of infeasibility of (12) if no solution exists. Now notice that (12) is a homogeneous conic feasibility problem of the form (5), as it is comprised of a single second-order cone inclusion constraint ($(t\|x\|A^*w, \langle w, -Ax\rangle) \in Q^n$) plus a constraint that the variable w must lie in K^*. Therefore, using Remark 2 and the premise that K^* has an efficiently-computable deep-separation oracle, it follows that (12) itself can be efficiently solved by the re-scaled perceptron algorithm, under the proviso that it has a solution.

However, in the case when (12) has no solution, it will be necessary to develop a means to certify this infeasibility. To do so, we first analyze its feasibility cone, denoted as $\tilde{\mathcal{F}}_{(t,x)} := \{w : t\|x\|\|A^*w\| + \langle w, Ax\rangle \le 0,\ w \in K^*\}$. We have:

Proposition 1. *For a given $\sigma \in (0, 1/2)$ and $x \ne 0$, suppose that $S_{(\sigma,x)}$ has a solution and let $t \in (0, \sigma)$. Then*

$$\tau_{\tilde{\mathcal{F}}_{(t,x)}} \ge \frac{\tau_{K^*}(\sigma - t)}{3\mathcal{C}(A)} \ .$$

Proof. For simplicity we assume with no loss of generality that $\|x\| = 1$ and $\|A\| = 1$. Since $S_{(\sigma,x)}$ has a solution, let $\hat{w}$ satisfy $\sigma\|A^*\hat{w}\| + \langle \hat{w}, Ax\rangle \le 0$, $\hat{w} \in K^*$, and $\|\hat{w}\| = 1$. It follows directly from Theorem 2 of [7] that $\|A^*\hat{w}\| \ge \rho(A)$. Let w° be the center of K^*, whereby $B(w^\circ, \tau_{K^*}) \subset K^*$. Consider the vector $\hat{w} + \beta w^\circ + \alpha d$ where $\|d\| \le 1$ and $\beta > 0$ will be specified shortly. Then $\hat{w} + \beta w^\circ + \alpha d \in K^*$ so long as $\alpha \le \beta\tau_{K^*}$. Also,

$$\begin{aligned} t\|A^*(\hat{w} + \beta w^\circ + \alpha d)\| + \langle \hat{w} + \beta w^\circ + \alpha d, Ax\rangle &\le t\|A^*\hat{w}\| + \beta t + \alpha t + \langle \hat{w}, Ax\rangle + \beta + \alpha \\ &\le (t - \sigma)\|A^*\hat{w}\| + \beta t + \alpha t + \beta + \alpha \\ &\le (t - \sigma)\rho(A) + \beta t + \alpha t + \beta + \alpha \\ &\le 0 \end{aligned}$$

so long as $\alpha \le \hat{\alpha} := \frac{(\sigma - t)\rho(A)}{t+1} - \beta$. Therefore

$$\tau_{\tilde{\mathcal{F}}_{(t,x)}} \ge \frac{\min\left\{\frac{(\sigma-t)\rho(A)}{t+1} - \beta,\ \beta\tau_{K^*}\right\}}{\|\hat{w} + \beta w^\circ\|} \ge \frac{\min\left\{\frac{(\sigma-t)\rho(A)}{t+1} - \beta,\ \beta\tau_{K^*}\right\}}{1 + \beta} \ .$$

Let $\beta := \frac{(\sigma-t)\rho(A)}{2(t+1)}$ and substituting in this last expression yields

$$\tau_{\tilde{\mathcal{F}}_{(t,x)}} \ge \frac{(\sigma - t)\rho(A)\tau_{K^*}}{2 + 2t + (\sigma - t)\rho(A)} \ge \frac{(\sigma - t)\rho(A)\tau_{K^*}}{3} = \frac{(\sigma - t)\tau_{K^*}}{3\mathcal{C}(A)}$$

since $\rho(A) \le \|A\| = 1$ and $0 < t \le \sigma \le 1/2$.

Now consider the following half-deep-separation oracle for $\mathcal{F}$ (recall Definition 4) which takes as input an estimate L of $\mathcal{C}(A)$:

Probabilistic Half-deep-separation Oracle for $\mathcal{F}$, for $x \neq 0$, relaxation parameter σ, failure probability δ, and estimate L

```
Set $t := \sigma/2$, and run the re-scaled perceptron algorithm to compute a solution $\tilde{w}$ of (12) for at most $\hat{T} := \max\left\{4096\ln\left(\frac{1}{\delta}\right), 139n\ln\left(\frac{6L}{\tau_{K^*}}\right)\right\}$ iterations.
  If a solution $\tilde{w}$ of (12) is computed, return $d := A^*\tilde{w}/\|A^*\tilde{w}\|$,
      report Condition II, and Stop.
  If no solution is computed within $\hat{T}$ iterations, report
      "either Condition I is satisfied, or $L < \mathcal{C}(A)$," and Stop.
```

The following states the correctness of the above oracle:

Theorem 2. *Using the iteration count $\hat{T}$ above, with probability at least $1 - \delta$ the output of the probabilistic oracle is correct.*

Proof. If the oracle computes a solution $\tilde{w}$ of (12), then it is trivial to show that $d := A^*\tilde{w}/\|A^*\tilde{w}\|$ satisfies $d \in \mathcal{F}^*$ and $\frac{\langle d,x\rangle}{\|d\|\|x\|} \leq -t = -\sigma/2$, thus satisfying condition II of Definition 4. Suppose instead that the oracle does not compute a solution within $\hat{T}$ iterations. If (12) has a solution it follows from Theorem 1 that with probability at least $1-\delta$ the re-scaled perceptron algorithm would compute a solution of (12) in at most

$$T := \max\left\{4096\ln\left(\frac{1}{\delta}\right), 139n\ln\left(\frac{1}{32n\tau_{\tilde{\mathcal{F}}_{(t,x)}}}\right)\right\}$$

iterations. However, if $L \geq \mathcal{C}(A)$ and $\tilde{\mathcal{F}}_{(\sigma,x)} \neq \emptyset$, then it follows from Proposition 1 that

$$\frac{1}{32n\tau_{\tilde{\mathcal{F}}_{(t,x)}}} \leq \frac{3\mathcal{C}(A)}{32n\tau_{K^*}(\sigma/2)} \leq \frac{6L}{\tau_{K^*}} ,$$

whereby $T \leq \hat{T}$. Therefore, it follows that that either $L < \mathcal{C}(A)$ or $\tilde{\mathcal{F}}_{(\sigma,x)} = \emptyset$, the latter then implying Condition I of Definition 4.

We note that the above-outlined method for constructing a deep-separation oracle is inelegant in many respects. Nevertheless, it is theoretically efficient, i.e., it is polynomial-time in n, $\ln(1/\tau_{K^*})$, $\ln(L)$, and $\ln(1/\delta)$. It is an interesting and open question whether, in the case of $K = S^{k\times k}_+$, a more straightforward and more efficient deep-separation oracle for $\mathcal{F}$ can be constructed.

Finally, it follows from Theorem 7 of [7] that the width of $\mathcal{F}$ can be lower-bounded by Renegar's condition measure:

$$\tau_{\mathcal{F}} \geq \frac{\tau_K}{\mathcal{C}(A)} . \tag{13}$$

This can be used in combination with binary search (for bounding $\mathcal{C}(A)$) and the half-deep-separation oracle above to produce a complexity bound for computing a solution of (1) in time polynomial in n, $\ln(\mathcal{C}(A))$, $\ln(1/\delta)$, $\ln(1/\tau_K)$, and $\ln(1/\tau_{K^*})$.

References

1. Alexandre Belloni and Robert M. Freund, *Characterization of second-order cone inclusion, and efficient projection*, Technical Report, MIT Operations Research Center, ORzzz-06 2006.
2. D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, Journal of the ACM **51** (2004), no. 4, 540–556.
3. A. Blum, A. Frieze, R. Kannan, and S. Vempala, *A polynomial-time algorithm for learning noisy linear threashold functions*, Algorithmica **22** (1998), no. 1, 35–52.
4. T. Bylander, *Learning linear threshold functions in the presence of classification noise*, Proceedings of the Seventh Annual Workshop on Computational Learning Theory (1994), 340–347.
5. J. Dunagan and S. Vempala, *A simple polynomial-time rescaling algorithm for solving linear programs*, Proceedings of STOC'04 (2004).
6. R. M. Freund and J. R. Vera, *Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm*, SIAM Journal on Optimization **10** (1999), no. 1, 155–176.
7. ———, *Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system*, Mathematical Programming **86** (1999), no. 2, 225–260.
8. Y. Freund and R. E. Schapire, *Large margin classification using the perceptron algorithm*, Machine Learning **37** (1999), no. 3, 297–336.
9. N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984), no. 4, 373–395.
10. L.G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl. **20** (1979), no. 1, 191–194.
11. M. Minsky and S. Papert, *Perceptrons: An introduction to computational geometry*, The MIT Press, 1969.
12. Y. Nesterov and A Nemirovskii, *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1993.
13. A. B. J. Novikoff, *On convergence proofs on perceptrons*, Proceedings of the Symposium on the Mathematical Theory of Automata **X11** (1962), 615–622.
14. G. Pataki, *On the closedness of the linear image of a closed convex cone*, Technical Report, University of North Carolina, TR-02-3 Department of Operations Research 1992.
15. J. Renegar, *Some perturbation theory for linear programming*, Mathematical Programming **65** (1994), no. 1, 73–91.
16. ———, *Linear programming, complexity theory, and elementary functional analysis*, Mathematical Programming **70** (1995), no. 3, 279–351.
17. F. Rosenblatt, *Principles of neurodynamics*, Spartan Books, Washington, DC, 1962.
18. H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of semidefinite programming*, Kluwer Academic Publishers, 2000.