A Modified Perceptron Algorithm

Javier Peña
Carnegie Mellon University

(joint work with Negar Soheili)

Fields Institute
September 2011

Perceptron Algorithm

Algorithm to solve

$$A^\mathsf{T} y > 0,$$

for a given $A := \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m\times n}$.

Perceptron Algorithm (Rosenblatt, 1958)

- $y := 0$
- while $A^\mathsf{T} y \not> 0$

  $y := y + \frac{a_j}{\|a_j\|}, \quad$ where $\; a_j^\mathsf{T} y \leq 0$

  end while

Throughout this talk: $\|\cdot\| = \|\cdot\|_2$.

Perceptron Algorithm

Attractive features of the Perceptron Algorithm

- Simple greedy iterations
- Simple convergence analysis (Block-Novikoff, 1962): Algorithm terminates in at most $\frac{1}{\rho(A)^2}$ iterations where

$$\rho(A) = \text{ thickness of } \{y : A^\mathsf{T} y \geq 0\}.$$

- Dunagan & Vempala 2004: Randomized re-scaled version that terminates in $\mathcal{O}\left(n \log\left(\frac{1}{\rho(A)}\right)\right)$ iterations with high probability.
- Belloni, Freund & Vempala 2007: Randomized re-scaled perceptron for general conic systems with similar convergence.

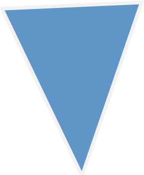
Thickness parameter $\rho(A)$

Assume

- $A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$, where $\|a_j\| = 1,\ j = 1, \ldots, n$.
- The problem $A^\mathsf{T} y > 0$ is feasible.

Definition

$$\begin{aligned}\rho(A) &= \max_{\|y\|=1} \left\{ r : \mathbb{B}(y,r) \subseteq \{z : A^\mathsf{T} z \geq 0\} \right\} \\ &= \max_{\|y\|=1} \min_i a_i^\mathsf{T} y.\end{aligned}$$



large $\rho(A)$

small $\rho(A)$

Main Theorem

Theorem (Soheili & P, 2011)

Modified version of the perceptron algorithm that terminates in $\mathcal{O}(\frac{\sqrt{\log(n)}}{\rho(A)})$ *iterations.*

Main Theorem

Theorem (Soheili & P, 2011)

Modified version of the perceptron algorithm that terminates in $\mathcal{O}(\frac{\sqrt{\log(n)}}{\rho(A)})$ *iterations.*

Remarks

- The modified version retains some/most of the algorithm's original simplicity.
- Unlike Dunagan and Vempala's, our algorithm is deterministic.
- Our iteration bound is weaker on $\rho(A)$ but stronger on n.

Classical Perceptron Algorithm

Classical Perceptron Algorithm

- $y_0 := 0$
- For $k = 0, 1, \ldots$

  $a_j^\mathsf{T} y_k := \min_i a_i^\mathsf{T} y_k$

  $y_{k+1} := y_k + a_j$

  end for

Classical Perceptron Algorithm

Classical Perceptron Algorithm

- $y_0 := 0$
- For $k = 0, 1, \ldots$

  $a_j^\mathsf{T} y_k := \min_i a_i^\mathsf{T} y_k$

  $y_{k+1} := y_k + a_j$

  end for

Observe

$$a_j^\mathsf{T} y := \min_i a_i^\mathsf{T} y \Leftrightarrow a_j = Ax(y),\ x(y) = \underset{x \in \Delta_n}{\operatorname{argmin}} \langle A^\mathsf{T} y, x \rangle.$$

Hence in the above algorithm $y_k = Ax_k$ where $x_k \geq 0, \ \ \|x_k\|_1 = k$.

Normalized Perceptron Algorithm

Recall $x(y) := \underset{x \in \Delta_n}{\operatorname{argmin}} \langle A^{\mathsf{T}} y, x \rangle$.

Normalized Perceptron Algorithm

- $y_0 := 0$
- For $k = 0, 1, \ldots$
  $\theta_k := \frac{1}{k+1}$
  $y_{k+1} := (1 - \theta_k) y_k + \theta_k A x(y_k)$

  end for

In this algorithm $y_k = A x_k$ for $x_k \in \Delta_n$.

Modified Perceptron Algorithm

Key step: Use a smooth version of

$$x(y) = \operatorname*{argmin}_{x \in \Delta_n} \langle A^{\mathsf{T}} y, x \rangle,$$

namely,

$$x_\mu(y) := \frac{\exp(-A^{\mathsf{T}} y/\mu)}{\|\exp(-A^{\mathsf{T}} y/\mu)\|_1}$$

for some $\mu > 0$.

Smooth Perceptron Algorithm

Smooth Perceptron Algorithm

- $y_0 := \frac{1}{n}A\mathbf{1}$; $\mu_0 := 1$; $x_0 := x_{\mu_0}(y_0)$
- for $k = 0, 1, \ldots$

  $\theta_k := \frac{2}{k+3}$

  $y_{k+1} := (1-\theta_k)(y_k + \theta_k A x_k) + \theta_k^2 A x_{\mu_k}(y_k)$

  $\mu_{k+1} := (1-\theta_k)\mu_k$

  $x_{k+1} := (1-\theta_k)x_k + \theta_k x_{\mu_{k+1}}(y_{k+1})$

  end for

Smooth Perceptron Algorithm

Smooth Perceptron Algorithm

- $y_0 := \frac{1}{n}A\mathbf{1}$; $\mu_0 := 1$; $x_0 := x_{\mu_0}(y_0)$
- for $k = 0, 1, \ldots$

  $\theta_k := \frac{2}{k+3}$

  $y_{k+1} := (1-\theta_k)(y_k + \theta_k A x_k) + \theta_k^2 A x_{\mu_k}(y_k)$

  $\mu_{k+1} := (1-\theta_k)\mu_k$

  $x_{k+1} := (1-\theta_k)x_k + \theta_k x_{\mu_{k+1}}(y_{k+1})$

  end for

Main loop in the normalized version:

for $k = 0, 1, \ldots$

$\theta_k := \frac{1}{k+1}$

$y_{k+1} := (1-\theta_k)y_k + \theta_k A x(y_k)$

end for

Thickness parameter (again)

Recall our assumptions:

- $A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$, where $\|a_j\| = 1,\ j = 1, \ldots, n$.
- Problem $A^\mathsf{T} y > 0$ is feasible.

Thickness parameter

$$\begin{aligned}
\rho(A) &= \max_{\|y\|=1} \min_j a_j^\mathsf{T} y \\
&= \max_{\|y\|\leq 1} \min_j a_j^\mathsf{T} y \\
&= \max_{\|y\|\leq 1} \psi(y),
\end{aligned}$$

where

$$\psi(y) := \min_{x \in \Delta_n} \langle A^\mathsf{T} y, x \rangle.$$

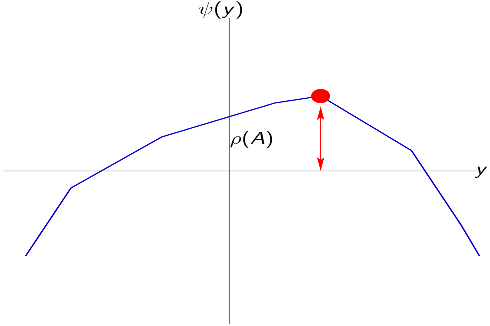
Thickness parameter (again)

We have

$$\rho(A) = \max_{||y||\leq 1} \psi(y).$$

Therefore, given $\|y\| \leq 1$

$$\begin{aligned} A^\mathsf{T} y > 0 &\Leftrightarrow \psi(y) > 0 \\ &\Leftrightarrow \psi(y) \text{ is within } \rho(A) \text{ of its max on } \{y : \|y\| \leq 1\}. \end{aligned}$$



Thickness parameter (again)

Similarly,

$$\frac{1}{2}\rho(A)^2 = \max_y \phi(y),$$

where

$$\phi(y) := -\frac{\|y\|^2}{2} + \min_{x \in \Delta_n} \langle A^\mathsf{T} y, x \rangle.$$

Furthermore, $A^\mathsf{T} y > 0$ if $\phi(y) > 0$.

Notice that $\phi(y) > 0 \Leftrightarrow \phi(y)$ is within $\frac{1}{2}\rho(A)^2$ of its maximum.

Perceptron Algorithm as a Subgradient Algorithm

Main loop in the Normalized Perceptron Algorithm:

for $k = 0, 1, \ldots$
  $\theta_k := \frac{1}{k+1}$
  $y_{k+1} := (1 - \theta_k) y_k + \theta_k A x(y_k) = y_k + \theta_k(-y_k + Ax(y_k))$

end for

Observe: $-y + Ax(y) \in \partial(-\phi)(y)$.

Perceptron Algorithm as a Subgradient Algorithm

Main loop in the Normalized Perceptron Algorithm:

for $k = 0, 1, \ldots$
  $\theta_k := \frac{1}{k+1}$
  $y_{k+1} := (1 - \theta_k) y_k + \theta_k A x(y_k) = y_k + \theta_k(-y_k + A x(y_k))$

end for

Observe: $-y + Ax(y) \in \partial(-\phi)(y)$.

Normalized Perceptron Algorithm

Subgradient algorithm for

$$\min_y (-\phi)(y) \;\Leftrightarrow\; \max_y \phi(y).$$

Smooth Perceptron Algorithm

Application of Excessive Gap Technique (Nesterov 2005).

Consider the maximization problem

$$\max_y \phi(y) = \max_y \min_{x \in \Delta_n} \left\{ -\frac{\|y\|^2}{2} + \langle A^\mathsf{T} y, x \rangle \right\}.$$

For $\mu > 0$ let

$$\phi_\mu(y) := -\frac{\|y\|^2}{2} + \min_{x \in \Delta_n} \{ \langle A^\mathsf{T} y, x \rangle + \mu d(x) \},$$

where $d(x) = \sum_{j=1}^{n} x_j \log(x_j) + \log(n).$

Smooth Perceptron Algorithm

Observe:

$$\phi_\mu(y) = -\frac{\|y\|^2}{2} + \langle A^\mathsf{T} y, x_\mu(y) \rangle + \mu d(x_\mu(y)),$$

where

$$x_\mu(y) = \frac{\exp(-A^\mathsf{T} y/\mu)}{\|\exp(-A^\mathsf{T} y/\mu)\|_1}.$$

Furthermore,

$$\nabla \phi_\mu(y) = -y + A x_\mu(y).$$

Main Theorem Again

Smooth Perceptron Algorithm

- $y_0 := \frac{1}{n}A\mathbf{1}$; $\mu_0 := 1$; $x_0 := x_{\mu_0}(y_0)$
- for $k = 0, 1, \ldots$

$$\theta_k := \frac{2}{k+3}$$
$$y_{k+1} := (1-\theta_k)(y_k + \theta_k A x_k) + \theta_k^2 A x_{\mu_k}(y_k)$$
$$\mu_{k+1} := (1-\theta_k)\mu_k$$
$$x_{k+1} := (1-\theta_k)x_k + \theta_k x_{\mu_{k+1}}(y_{k+1})$$

end for

Theorem (Soheili & P, 2011)

Smooth Perceptron Algorithm terminates in at most

$$\frac{2\sqrt{\log(n)}}{\rho(A)} - 1$$

iterations.

Proof of Main Theorem

Claim

For all $x \in \Delta_n$ we have $\rho(A) \leq \|Ax\|$.

Claim

For all $y \in \mathbb{R}^m$ we have $\phi_\mu(y) \leq \phi(y) + \mu \log(n)$.

Lemma

The iterates $x_k \in \Delta_n,\ y_k \in \mathbb{R}^m,\ k = 0, 1, \ldots$ generated by the Smooth Perceptron Algorithm satisfy the Excessive Gap Condition

$$\frac{1}{2}\|Ax_k\|^2 \leq \phi_{\mu_k}(y_k).$$

Proof of Main Theorem

Putting together the two claims and lemma we get

$$\frac{1}{2}\rho(A)^2 \leq \frac{1}{2}\|Ax_k\|^2 \leq \phi_{\mu_k}(y_k) \leq \phi(y_k) + \mu_k \log(n).$$

So

$$\phi(y_k) \geq \frac{1}{2}\rho(A)^2 - \mu_k \log(n).$$

In the algorithm $\mu_k = 1 \cdot \frac{1}{3} \cdot \frac{2}{4} \cdots \frac{k}{k+2} = \frac{2}{(k+1)(k+2)} < \frac{2}{(k+1)^2}$.

Thus $\phi(y_k) > 0$, and consequently $A^\mathsf{T} y_k > 0$, as soon as

$$k \geq \frac{2\sqrt{\log(n)}}{\rho(A)} - 1.$$

□

Proof of Lemma

We need to show

$$\frac{1}{2}\|Ax_k\|^2 \leq \phi_{\mu_k}(y_k).$$

The proof uses two key ingredients.

- Bregman distance: For $z, x \in \Delta_n$

$$h(z,x) := d(z) - d(x) - \langle \nabla d(x), z - x \rangle \geq \frac{1}{2}\|z - x\|_1^2.$$

- $(2,1)$-norm of A

$$\begin{aligned} \|A\|_{2,1} &= \max_{\|x\|_1=1} \|Ax\| \\ &= \max\{\|a_1\|, \ldots, \|a_n\|\} = 1. \end{aligned}$$

Proof of Lemma

$k = 0$:

$$\begin{array}{rcl} \frac{1}{2}\|Ax_0\|^2 & = & \frac{1}{2}\|A\frac{\mathbf{1}}{n}\|^2 + \langle A\frac{\mathbf{1}}{n}, A(x_0 - \frac{\mathbf{1}}{n})\rangle + \|A(x_0 - \frac{\mathbf{1}}{n})\|^2 \\ & \leq & -\frac{1}{2}\|y_0\|^2 + \langle A^\mathsf{T} y_0, x_0\rangle + \frac{1}{2}\left\|x_0 - \frac{\mathbf{1}}{n}\right\|_1^2 \\ & \leq & -\frac{1}{2}\|y_0\|^2 + \langle A^\mathsf{T} y_0, x_{\mu_0}(y_0)\rangle + d(x_{\mu_0}(y_0)) \\ & = & \phi_{\mu_0}(y_0). \end{array}$$

$k \Rightarrow k+1$: To ease notation drop k, put $\hat{x} = (1-\theta)x + \theta x_\mu(y)$. Hence $y_+ = (1-\theta)y + \theta A\hat{x}$ and $x_+ = (1-\theta)x + \theta x_{\mu_+}(y_+)$.

$$\begin{array}{rcl} \phi_{\mu_+}(y_+) & = & -\frac{1}{2}\|y_+\|^2 + \langle A^\mathsf{T} y_+, x_{\mu_+}\rangle + \mu_+ d(x_{\mu_+}) \\ & \geq & (1-\theta)\left[-\frac{1}{2}\|y\|^2 + \langle A^\mathsf{T} y, x_{\mu_+}\rangle + \mu d(x_{\mu_+})\right] + \\ & & +\theta\left[-\frac{1}{2}\|A\hat{x}\|^2 + \langle A^\mathsf{T} A\hat{x}, x_{\mu_+}\rangle\right]. \end{array}$$

Proof of Lemma

Next, observe that

$$
\begin{aligned}
&-\tfrac{1}{2}\|y\|^2 + \langle A^{\mathsf T} y, x_{\mu_+}\rangle + \mu d(x_{\mu_+}) \\
&= -\tfrac{1}{2}\|y\|^2 + \langle A^{\mathsf T} y, x_{\mu}\rangle + \mu d(x_{\mu}) + \mu(d(x_{\mu_+}) - d(x_\mu) - \langle \nabla d(x_\mu), x_{\mu_+} - x_\mu\rangle) \\
&= \phi_\mu(y) + \mu(d(x_{\mu_+}) - d(x_\mu) - \langle \nabla d(x_\mu), x_{\mu_+} - x_\mu\rangle) \\
&\geq \tfrac{1}{2}\|Ax\|^2 + \tfrac{1}{2}\mu\|x_{\mu_+} - x_\mu\|_1^2 \\
&\geq \tfrac{1}{2}\|A\hat{x}\|^2 + \langle A^{\mathsf T} A\hat{x}, x - \hat{x}\rangle + \tfrac{1}{2}\mu\|x_{\mu_+} - x_\mu\|_1^2,
\end{aligned}
$$

and

$$
-\frac{1}{2}\|A\hat{x}\|^2 + \langle A^{\mathsf T} A\hat{x}, x_{\mu_+}\rangle = \frac{1}{2}\|A\hat{x}\|^2 + \langle A^{\mathsf T} A\hat{x}, x_{\mu_+} - \hat{x}\rangle.
$$

At iteration k we have $\frac{1}{2}(1-\theta)\mu = \frac{2}{(k+2)(k+3)} < \frac{2}{(k+3)^2} = \frac{1}{2}\theta^2$. Therefore

$$
\begin{aligned}
\phi_{\mu_+}(y_+) &\geq \tfrac{1}{2}\|A\hat{x}\|^2 + \theta\langle A^{\mathsf T} A\hat{x}, x_{\mu_+} - x_\mu\rangle + \tfrac{1}{2}\theta^2\|x_{\mu_+} - x_\mu\|_1^2 \\
&\geq \tfrac{1}{2}\|A\hat{x}\|^2 + \langle A^{\mathsf T} A\hat{x}, x_+ - \hat{x}\rangle + \tfrac{1}{2}\|x_+ - \hat{x}\|_1^2 \\
&\geq \tfrac{1}{2}\|Ax_+\|^2.
\end{aligned}
$$

□

Numerical Experiments

Recall:

	Classical Perceptron	Smooth Perceptron
Complexity	$\frac{1}{\rho(A)^2}$	$\frac{2\sqrt{\log(n)}}{\rho(A)} - 1$

This suggests relationship:

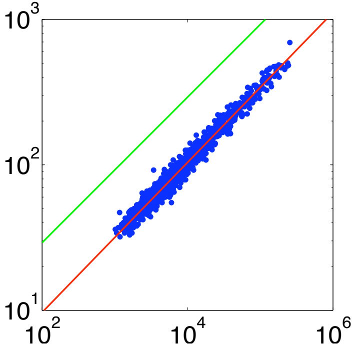
$$Y = 2\sqrt{\log(n) \cdot X}$$

between

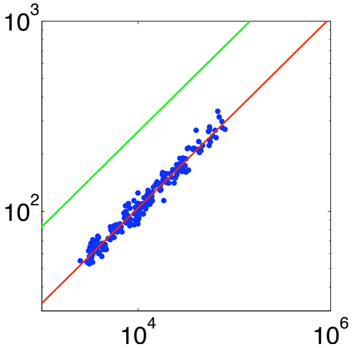
Y = number of iterations in Smooth Perceptron algorithm

X = number iterations in Classical Perceptron algorithm.

Number of iterations for randomly generated instances

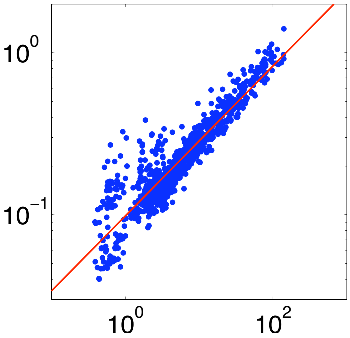


$A \in \mathbb{R}^{100\times 500}$

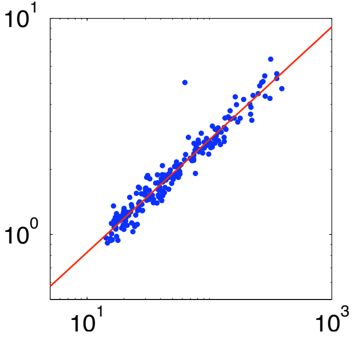


$A \in \mathbb{R}^{200\times 1000}$

CPU times for randomly generated instances



$A \in \mathbb{R}^{100\times 500}$



$A \in \mathbb{R}^{200\times 1000}$

What if $A^\mathsf{T} y > 0$ is infeasible?

In this case the alternative

$$Ax = 0,\ x \in \Delta_n$$

is feasible and $\rho(A) = \max\limits_{\|y\|=1} \min\limits_i a_i^\mathsf{T} y \leq 0$.

Ill-posedness

A is **ill-posed** when $\rho(A) = 0$. In this case both $A^\mathsf{T} y > 0$ and $Ax = 0, x > 0$ are on the verge of feasibility.

Theorem (Cheung & Cucker, 2001)

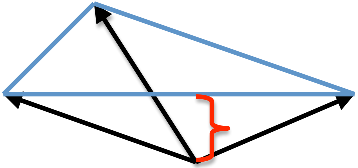
$$|\rho(A)| = \min\{\|\tilde{A} - A\|_{2,1} : \tilde{A} \textit{ is ill-posed}\}.$$

We continue to assume $A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix},\ \|a_i\| = 1,\ i = 1, \ldots, n.$

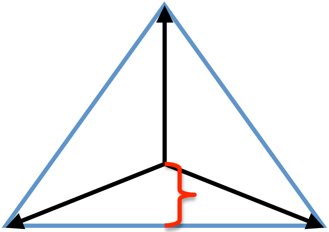
Some geometry

Proposition (From Renegar 1995 and Cheung-Cucker 2001)

$$|\rho(A)| = \min\left\{\|Ax\| : x \geq 0,\ x \in \Delta_n\right\}.$$



$\rho(A) > 0$



$\rho(A) < 0$

What about the perceptron algorithm?

Assume $A^{\mathsf{T}} y > 0$ is infeasible.

Theorem (Dantzig, 1992)

The iterates $x_k \in \Delta_n$ generated by the normalized perceptron satisfy

$$\|Ax_k\| \leq \frac{1}{\sqrt{k}}.$$

What about the perceptron algorithm?

Assume $A^{\mathsf{T}} y > 0$ is infeasible.

Theorem (Dantzig, 1992)

The iterates $x_k \in \Delta_n$ generated by the normalized perceptron satisfy

$$\|Ax_k\| \leq \frac{1}{\sqrt{k}}.$$

Proposition (Soheili & P, 2011)

The iterates $x_k \in \Delta_n$ generated by the smooth perceptron satisfy

$$\|Ax_k\| \leq \frac{2\sqrt{\log(n)}}{k+1}.$$

Von Neumann Algorithm

Algorithm to solve

$$Ax = 0,\ x \in \Delta_n. \tag{1}$$

Von Neumann Algorithm, 1948

- $x_0 := \frac{1}{n}\mathbf{1}$; $y_0 := Ax_0$
- For $k = 0, 1, \ldots$

  if $v_k := \min_i a_i^{\mathsf{T}} y_k > 0$ then STOP; (1) is infeasible

  $\lambda_k := \frac{1 - v_k}{\|y_k\|^2 - 2v_k + 1}$

  $x_{k+1} := \lambda_k x_k + (1 - \lambda_k)x(y_k)$

  end for

Von Neumann Algorithm

Algorithm to solve

$$Ax = 0, \; x \in \Delta_n. \tag{1}$$

Von Neumann Algorithm, 1948

- $x_0 := \frac{1}{n}\mathbf{1}$; $y_0 := Ax_0$
- For $k = 0, 1, \ldots$

  if $v_k := \min_i a_i^\mathsf{T} y_k > 0$ then STOP; (1) is infeasible

  $\lambda_k := \frac{1-v_k}{\|y_k\|^2 - 2v_k + 1}$

  $x_{k+1} := \lambda_k x_k + (1-\lambda_k)x(y_k)$

  end for

Main loop in the normalized perceptron:

for $k = 0, 1, \ldots$

$\theta_k := \frac{1}{k+1}$

$x_{k+1} := (1-\theta_k)x_k + \theta_k x(y_k)$

end for

Von Neumann Algorithm

Theorem (Dantzig, 1992)

If (1) *is feasible, then the Von Neumann Algorithm finds an* ϵ*-solution to* (1) *in at most*

$$\frac{1}{\epsilon^2}$$

iterations.

Von Neumann Algorithm

Theorem (Dantzig, 1992)

If (1) *is feasible, then the Von Neumann Algorithm finds an* ϵ*-solution to* (1) *in at most*

$$\frac{1}{\epsilon^2}$$

iterations.

Theorem (Epelman & Freund, 2000)

If (1) *is feasible and* $\rho(A) < 0$*, then the Von Neumann Algorithm finds an* ϵ*-solution to* (1) *in at most*

$$\frac{1}{\rho(A)^2} \cdot \log\left(\frac{1}{\epsilon}\right)$$

iterations.

Conjecture

If $Ax = 0,\ x \in \Delta_n$ is feasible and $\rho(A) < 0$ then the smooth perceptron, or a suitable smooth Von Neumann algorithm, finds an ϵ-solution in

$$\mathcal{O}\left(\frac{\sqrt{\log(n)}}{|\rho(A)|} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$$

iterations.

Conclusion

- Smooth perceptron algorithm improves complexity from $\mathcal{O}(\frac{1}{\rho(A)^2})$ to $\mathcal{O}(\frac{\sqrt{\log(n)}}{\rho(A)})$.
- Modification preserves some/most of the algorithm's original simplicity.
- Current & future work:
  – Smooth Von Neumann Algorithm
  – General conic systems (in Belloni-Freund-Vempala's spirit).