

On the complexity of elementary algorithms for convex optimization

Javier Peña
Carnegie Mellon University

(joint work with Negar Soheili)

UN Encuentro de Matemáticas
Universidad Nacional
July 2012

Preamble

Convex optimization:

$$\min_{x \in Q} f(x)$$

f convex function, Q convex set.

Main algorithmic approaches

- First-order methods: gradient or subgradient descent.
- Second-order methods: Newton's method.
- First-order algorithms currently dominate research in large-scale convex optimization.

Theme

- Complexity analysis of first-order algorithms.
- Concentrate on two classical elementary algorithms for linear programming: The *perceptron* and *von Neumann's* algorithms.

Perceptron Algorithm

Algorithm to solve

$$A^T y > 0,$$

for a given $A := [a_1 \ a_2 \ \cdots \ a_n] \in \mathbb{R}^{m \times n}$.

Perceptron Algorithm (Rosenblatt, 1958)

- $y := 0$
- while $A^T y \not> 0$
 - $y := y + \frac{a_j}{\|a_j\|}$, where $a_j^T y \leq 0$
- end while

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

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

- Dunagan & Vempala 2004: Randomized re-scaled version that terminates in $\mathcal{O}\left(n^3 \log\left(\frac{1}{\rho(A)}\right)\right)$ elementary iterations with high probability.
- Belloni, Freund & Vempala 2007: Randomized re-scaled perceptron for general conic systems with similar convergence.

Thickness parameter $\rho(A)$

Assume

- $A = [a_1 \ \cdots \ a_n]$, where $\|a_j\| = 1$, $j = 1, \dots, n$.
- The problem $A^T y > 0$ is feasible.

Definition

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



large $\rho(A)$



small $\rho(A)$

Main Theorem

Theorem (Soheili & P, 2011)

Smooth perceptron algorithm that terminates in at most

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

elementary iterations.

Remarks

- Smooth version retains 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 and involves no big constants.
- Smooth perceptron for general conic systems $A^T y \in K$.

Classical Perceptron Algorithm

Classical Perceptron Algorithm

- $y_0 := 0$
 - for $k = 0, 1, \dots$
 - $a_j^T y_k := \min_i a_i^T y_k$
 - $y_{k+1} := y_k + a_j$
- end for

Observe

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

where $\Delta_n := \{x \in \mathbb{R}_+^n : \|x\|_1 = 1\}$.

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

Normalized Perceptron Algorithm

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

Normalized Perceptron Algorithm

- $y_0 := 0$
 - for $k = 0, 1, \dots$
 - $\theta_k := \frac{1}{k+1}$
 - $y_{k+1} := (1 - \theta_k)y_k + \theta_k Ax(y_k)$
- end for

In this algorithm $y_k = Ax_k$ for $x_k \in \Delta_n = \{x \in \mathbb{R}_+^n : \|x\|_1 = 1\}$.

Smooth Perceptron Algorithm

Key step

Use a smooth version of

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

namely,

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

for some $\mu > 0$.

Smooth Perceptron Algorithm

Smooth Perceptron Algorithm

- $y_0 := \frac{1}{n}A\mathbf{1}$; $\mu_0 := 2$; $x_0 := x_{\mu_0}(y_0)$
- for $k = 0, 1, \dots$
 - $\theta_k := \frac{2}{k+3}$
 - $y_{k+1} := (1 - \theta_k)(y_k + \theta_k Ax_k) + \theta_k^2 Ax_{\mu_k}(y_k)$
 - $\mu_{k+1} := (1 - \theta_k)\mu_k$
 - $x_{k+1} := (1 - \theta_k)x_k + \theta_k x_{\mu_{k+1}}(y_{k+1})$

end for

Main loop in the normalized version:

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

end for

Perceptron algorithm as a subgradient algorithm

Let

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

Observe

$$\max_y \phi(y) = \min_{x \in \Delta_n} \frac{1}{2} \|Ax\|^2 = \frac{1}{2} \rho(A)^2.$$

Perceptron update:

$$y_{k+1} = y_k + \theta_k (-y_k + Ax(y_k))$$

is precisely a subgradient update for

$$\max_y \phi(y).$$

Smooth perceptron algorithm as a gradient algorithm

Recall

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

Let the *smooth* approximation ϕ_μ of ϕ be defined as

$$\begin{aligned} \phi_\mu(y) &:= -\frac{\|y\|^2}{2} + \min_{x \in \Delta_n} \left\{ \langle A^\top y, x \rangle + \mu d(x) \right\} \\ &= -\frac{\|y\|^2}{2} + \langle A^\top y, x_\mu(y) \rangle + \mu d(x_\mu(y)), \end{aligned}$$

where $\mu > 0$ and $d(x) = \sum_{j=1}^n x_j \log(x_j) + \log(n)$.

Smooth perceptron: gradient scheme for $\max_y \phi_\mu(y)$.

Proof of Main Theorem

Apply Nesterov's excessive gap technique (Nesterov, 2005).

Claim

For all $x \in \Delta_n$ and $y \in \mathbb{R}^m$ we have $\phi(y) \leq \frac{1}{2} \|Ax\|^2$.

Claim

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

Lemma

The iterates $x_k \in \Delta_n$, $y_k \in \mathbb{R}^m$, $k = 0, 1, \dots$ 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 $\mu_k = 2 \cdot \frac{1}{3} \cdot \frac{2}{4} \cdots \frac{k}{k+2} = \frac{4}{(k+1)(k+2)} < \frac{4}{(k+1)^2}$.

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

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



Numerical Experiments

Recall:

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

This suggests relationship:

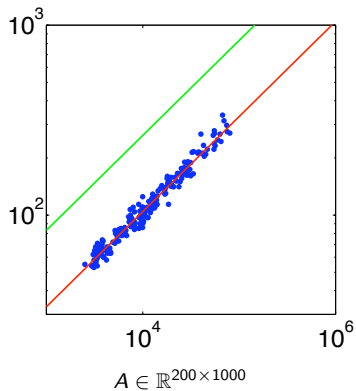
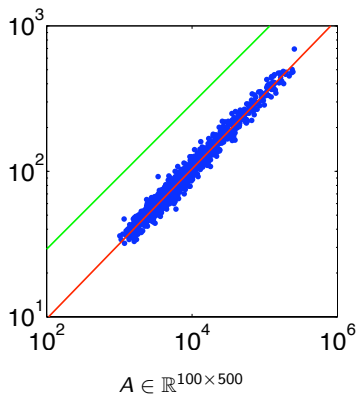
$$Y = 2\sqrt{2 \log(n)} \cdot X$$

between

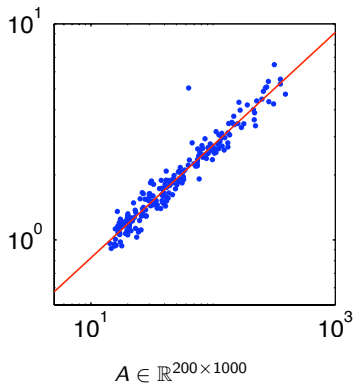
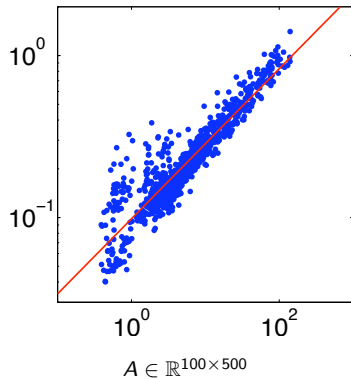
Y = number of iterations in Smooth Perceptron algorithm

X = number iterations in Classical Perceptron algorithm.

Number of iterations for randomly generated instances



CPU times for randomly generated instances



General conic feasibility problem

Assume $K \subseteq \mathbb{R}^n$ is a regular closed convex cone with dual K^* .

Given $A \in \mathbb{R}^{m \times n}$, consider the problem

$$A^T y \in \text{int}(K^*).$$

Cone of feasible solutions $\mathcal{F} := \{y : A^T y \in K^*\}$.

Interior separation oracle for \mathcal{F}

If $A^T y \notin \text{int}(K^*)$ find $u \in \mathcal{F}^*$, $u \neq 0$ such that $\langle u, y \rangle \leq 0$.

Assume

- The problem $A^T y \in \text{int}(K^*)$ is feasible.
- An interior separation oracle for \mathcal{F} is available.
(This is the case if an interior separation oracle for K is available.)

General perceptron algorithm

General Perceptron Algorithm (Belloni et al, 2007)

- $y := 0$
- while $A^T y \notin \text{int}(K^*)$
 - $y := y + u$, where $u \in \mathcal{F}^*$, $\|u\| = 1$, and $u^T y \leq 0$
- end while

Thickness of cone \mathcal{F}

$$\tau_{\mathcal{F}} := \max_{\|y\|=1} \{r : \mathbb{B}(y, r) \subseteq \mathcal{F}\}.$$

Proposition (Belloni et al, 2007)

General perceptron algorithm terminates in at most $\frac{1}{\tau_{\mathcal{F}}^2}$ iterations.

Smooth perceptron algorithm for conic systems

We need something like Δ_n and $x_\mu(y)$ for general K .

Coefficient of linearity (Freund & Vera 1999)

$$\beta_K := \max_{\|u\|^*=1} \min_{x \in K, \|x\|=1} \langle u, x \rangle.$$

Since $K \subseteq \mathbb{R}^n$ is a regular cone:

- (i) $0 < \beta_K \leq 1$ and $\beta_K = 1$ for a canonical norm in \mathbb{R}^n .
- (ii) There exists $\mathbf{1} \in K^*$ such that $\|\mathbf{1}\|^* = 1$ and

$$\beta_K = \min_{x \in K, \|x\|=1} \langle \mathbf{1}, x \rangle.$$

Examples

In all of the following cases $\beta_K = 1$:

- $K = \mathbb{R}_+^n$, $\|x\| = \sum_{i=1}^n |x_i|$, $\mathbf{1} = [1 \ \dots \ 1]^T$.
- $K = \mathbb{S}_+^n$, $\|X\| = \sum_{i=1}^n \sigma_i(X)$, $\mathbf{1} = I_n$.
- $K = \mathcal{L}_n := \left\{ x = \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^n : x_0 \geq \|\bar{x}\|_2 \right\}$, $\|x\| = |x_0| + \|\bar{x}\|_2$,
 $\mathbf{1} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$.
- A cartesian product of cones of the above three types (with total norm = sum of individual norms).

Let $\Delta(K) := \{x \in K : \langle \mathbf{1}, x \rangle = 1\}$.

Assume

- There is an oracle that for any $g \in \mathbb{R}^n$ finds

$$\operatorname{argmin}_x \{\langle g, x \rangle + d(x) : x \in \Delta(K)\}$$

for a prox-function $d : \Delta(K) \rightarrow \mathbb{R}$.

- Assume d has strong convexity parameter 1 and min value 0.

Examples (for $K = \mathbb{S}_+^n$)

- $d(X) = \sum_{i=1}^n \lambda_i(X) \log(\lambda_i(X)) + \log(n)$
- $d(X) = \frac{1}{2} \operatorname{trace}(X^2) - \frac{1}{2n} = \frac{1}{2} \|X\|_F^2 - \frac{1}{2n}$

Smooth perceptron algorithm

Let $\bar{x} := \operatorname{argmin}_{x \in \Delta(K)} d(x)$.

For $\mu > 0$ let $x_\mu : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be defined as

$$x_\mu(y) := \operatorname{argmin}_x \left\{ \langle A^\top y, x \rangle + d(x) : x \in \Delta(K) \right\}.$$

Smooth Perceptron Algorithm

- $y_0 := A\bar{x}$; $\mu_0 := 2\|A\|^2$; $x_0 := x_{\mu_0}(y_0)$
- for $k = 0, 1, \dots$
 - $\theta_k := \frac{2}{k+3}$
 - $y_{k+1} := (1 - \theta_k)(y_k + \theta_k A x_k) + \theta_k^2 A x_{\mu_k}(y_k)$
 - $\mu_{k+1} := (1 - \theta_k)\mu_k$
 - $x_{k+1} := (1 - \theta_k)x_k + \theta_k x_{\mu_{k+1}}(y_{k+1})$

end for

Main Theorem (extended version)

Theorem (Soheili & P, 2012)

Smooth perceptron algorithm terminates in at most

$$\frac{2\|A\|\sqrt{2D}}{\rho(A)} - 1$$

elementary iterations.

Here $\rho(A) := \max_{\|y\|=1} \min_{x \in \Delta(K)} \langle A^T y, x \rangle$ and $D = \max_{x \in \Delta(K)} d(x)$.

Remarks

- General perceptron terminates in at most $\frac{1}{\tau_{\mathcal{F}}^2}$ iterations (Belloni et al 2007).
- Freund & Vera 1999 showed $\tau_{\mathcal{F}} \geq \frac{\beta_K \cdot \rho(A)}{\|A\|}$
- For $K = \mathbb{R}_+^n$ and properly scaled A , we have $\tau_{\mathcal{F}} = \frac{\rho(A)}{\|A\|}$.
- $C(A) := \frac{\|A\|}{\rho(A)}$ = condition number for $A^T y \in K$ (Renegar).

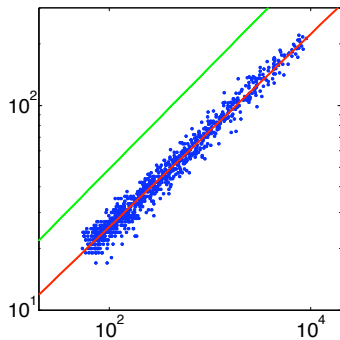
More numerical experiments

$K = \mathbb{S}_+^n$, $A : \mathbb{S}^n \rightarrow \mathbb{R}^m$ for randomly generated A .

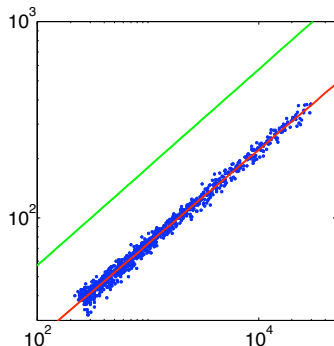
Let:

Y = number of iterations in Smooth Perceptron algorithm

X = number iterations in Classical Perceptron algorithm.



$m = 15, n = 20$



$m = 30, n = 60$

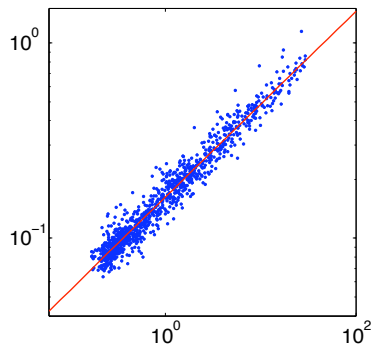
More numerical experiments

$K = \mathbb{S}_+^n$, $A : \mathbb{S}^n \rightarrow \mathbb{R}^m$ for randomly generated A .

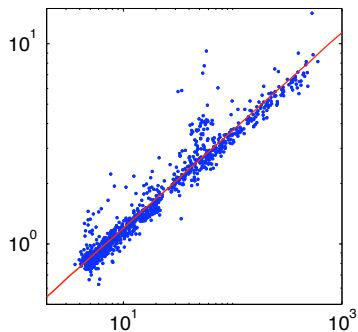
Let:

Y = CPU time taken by Smooth Perceptron algorithm

X = CPU time taken by Perceptron algorithm.



$m = 15, n = 20$



$m = 30, n = 60$

What if $A^T y \in \text{int}(K^*)$ is infeasible?

In this case the alternative

$$Ax = 0, x \in \Delta(K)$$

is feasible and

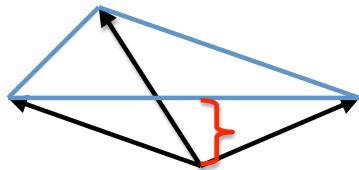
$$\rho(A) = \max_{\|y\|=1} \min_{x \in \Delta(K)} \langle A^T y, x \rangle \leq 0.$$

Recall

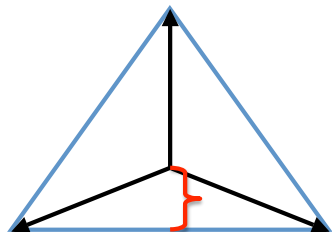
$$\Delta(K) = \{x \in K : \langle \mathbf{1}, x \rangle = 1\}.$$

Geometric interpretation

Blue set: $\{Ax : x \in \Delta(K)\}$.



$$\rho(A) > 0$$



$$\rho(A) < 0$$

Renegar's condition number

$$C(A) := \frac{\|A\|}{|\rho(A)|}.$$

Von Neumann Algorithm ($K = \mathbb{R}_+^n$)

Algorithm to solve

$$Ax = 0, x \in \Delta_n. \quad (1)$$

Von Neumann Algorithm, 1948

- $x_0 := \frac{1}{n}\mathbf{1}; y_0 := Ax_0$
- for $k = 0, 1, \dots$
 - if $v_k := \min_i a_i^\top y_k > 0$ then STOP; (1) is infeasible
 - $\lambda_k := \frac{1-v_k}{\|y_k\|^2 - 2v_k + 1}$
 - $x_{k+1} := \lambda_k x_k + (1 - \lambda_k)x(y_k)$
 - $y_{k+1} := \lambda_k y_k + (1 - \lambda_k)Ax(y_k)$end for

Main loop in the normalized perceptron:

- for $k = 0, 1, \dots$
 - $\theta_k := \frac{1}{k+1}$
 - $x_{k+1} := (1 - \theta_k)x_k + \theta_k x(y_k)$end for

Von Neumann Algorithm ($K = \mathbb{R}_+^n$)

Theorem (Dantzig, 1992)

If (1) is feasible, then the Von Neumann Algorithm finds an ϵ -solution to (1) in at most $\frac{\|A\|^2}{\epsilon^2}$ 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

$$C(A)^2 \cdot \log\left(\frac{\|A\|}{\epsilon}\right)$$

iterations.

Recall $C(A) = \|A\|/|\rho(A)|$.

Von Neumann Algorithm (general K)

Assume $K \subseteq \mathbb{R}^n$ is a regular closed convex cone with dual K^* .

Given $A \in \mathbb{R}^{m \times n}$, consider the alternative systems

$$A^T y \in \text{int}(K^*) \quad (\text{D})$$

and

$$Ax = 0, x \in \Delta(K). \quad (\text{P})$$

Assume

There is an oracle that for any $y \in \mathbb{R}^m$ finds

$$x(y) := \underset{x}{\operatorname{argmin}} \left\{ \langle A^T y, x \rangle : x \in \Delta(K) \right\}.$$

Von Neumann Algorithm (general K)

Von Neumann Algorithm (Epelman & Freund, 2000)

- $x_0 \in \Delta(K)$ arbitrary; $y_0 := Ax_0$
- for $k = 0, 1, \dots$
 - if $v_k := \min_{x \in \Delta(K)} \langle A^T y_k, x \rangle > 0$ then STOP; $A^T y_k \in \text{int}(K^*)$.
 - $\lambda_k := \frac{1 - v_k}{\|y_k\|^2 - 2v_k + 1}$
 - $x_{k+1} := \lambda_k x_k + (1 - \lambda_k)x(y_k)$
 - $y_{k+1} := \lambda_k y_k + (1 - \lambda_k)Ax(y_k)$

end for

Von Neumann Algorithm (general K)

Theorem (Epelman & Freund, 2000)

Assume $|\rho(A)| > 0$.

- (a) If $\rho(A) > 0$ (i.e., (D) is feasible), then von Neumann's Algorithm finds a solution to (D) in at most

$$C(A)^2$$

iterations.

- (b) If $\rho(A) > 0$ (i.e., (P) is feasible), then von Neumann's Algorithm finds an ϵ -solution to (P) in at most

$$C(A)^2 \cdot \log \left(\frac{\|A\|}{\epsilon} \right)$$

iterations.

Smooth Perceptron/von Neumann Algorithm

Theorem (Soheili & P, 2012)

Smooth Perceptron/von Neumann Algorithm such that:

(a) *If $\rho(A) > 0$, then algorithm finds a solution to (D) in at most*

$$\mathcal{O}(C(A) \cdot \log(C(A)))$$

elementary iterations.

(b) *If $\rho(A) < 0$, then algorithm finds an ϵ -solution to (P) in at most*

$$\mathcal{O}\left(C(A) \cdot \log\left(\frac{\|A\|}{\epsilon}\right)\right)$$

elementary iterations.

Summary

- Smooth versions of the perceptron and von Neumann's algorithm improve condition-based complexity roughly from $C(A)^2$ to $C(A)$.
- Smooth versions preserve most of the algorithms' original simplicity.
- Similar results are likely to hold for other first-order algorithms.