Applying Subgradient Methods
– and Accelerated Gradient Methods –
to Solve General Convex, Conic Optimization Problems

Jim Renegar
School of Operations Research
and Information Engineering
Cornell University

Workshop on “Optimization and Parsimonious Modeling”
IMA, January 25-29, 2016
convex function (assume lower semicontinuous)

\[
\min \ f(x) \quad \text{closed, convex set}
\]

\[
\text{s.t. } \ x \in Q
\]

The set of all subgradients at \(x\) is denoted \(\partial f(x)\) – the “subdifferential” at \(x\)
For a concave function, supgradients play the analogous role.
convex function (assume lower semicontinuous)

\[
\min \ f(x) \quad \text{closed, convex set} \\
\text{s.t.} \quad x \in Q
\]

Assume \( f \) is Lipschitz-continuous on an open neighborhood of \( Q \):

\[
|f(x) - f(y)| \leq M \|x - y\|
\]

Lipschitz constant

**Goal:** Compute \( x \in Q \) satisfying \( f(x) \leq f^* + \epsilon \)

A typical subgradient method:

**Initialize:** \( x_0 \in Q \)

**Iterate:** Compute \( g_k \in \partial f(x_k) \), and let \( x_{k+1} = P_Q(x_k - \frac{\epsilon}{\|g_k\|^2} g_k) \)

where \( P_Q \) is projection onto \( Q \)

A typical theorem:

\[
\ell \geq \left( \frac{M\|x_0 - x^*\|}{\epsilon} \right)^2 \quad \Rightarrow \quad \min_{k \leq \ell} f(x_k) \leq f^* + \epsilon
\]
In the special case of linear programming this becomes \( \ldots \)

\[
\begin{align*}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax \geq b \quad \text{so } Q = \{ x : Ax \geq b \}
\end{align*}
\]

Then, of course, the objective function is Lipschitz continuous:

\[
|c^T x - c^T y| \leq \|c\| \|x - y\|
\]

\( \text{Lipschitz constant} \)

**Goal:** Compute \( x \) satisfying \( Ax \geq b \) and \( c^T x \leq z^* + \epsilon \)

---

**A typical subgradient method:**

- **Initialize:** \( x_0 \in Q \)

- **Iterate:** Let \( x_{k+1} = P_Q( x_k - \frac{\epsilon}{\|c\|^2} c ) \)

  \( \text{where } P_Q \text{ is projection onto } Q \)

**A typical theorem:**

- \( \ell \geq \left( \frac{\|c\| \|x_0 - x^*\|}{\epsilon} \right)^2 \Rightarrow \min_{k \leq \ell} c^T x_k \leq z^* + \epsilon \)
But in general, projecting onto $Q = \{ x : Ax \geq b \}$ is no easier than solving linear programs!!!

A typical subgradient method:

Initialize: $x_0 \in Q$

Iterate: Let $x_{k+1} = P_Q(x_k - \frac{\epsilon}{\|c\|^2}c)$

where $P_Q$ is projection onto $Q$
**But in general, projecting onto** $Q = \{x : Ax \geq b\}$
**is no easier than solving linear programs!!!**

There are ways, however, to use a subgradient method to “solve” an LP.

For example, here is a way to “approximate” an LP by an unconstrained convex optimization problem:

$$\min \ c^T x \quad \text{s.t.} \quad Ax \geq b \quad \approx \quad \min \ c^T x + \gamma \max\{0, b_i - \alpha_i^T x : i = 1, \ldots, m\}$$

However, the optimal solution for the problem on the right will not necessarily be feasible for LP.

In the literature, in fact, the only subgradient methods producing feasible iterates require the feasible region to be “simple.”

“Why is this?”
Think of this 2-dimensional plane as being the slice of $\mathbb{R}^n$ cut out by $\{x : Ax = b\}$.
\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

\[
\pi^* \quad \text{optimal solution}
\]

Assume the objective function \( x \mapsto c^T x \) is constant on horizontal slices.
To begin with simplicity, assume the vector of all one’s is feasible.
\[
\begin{aligned}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{aligned}
\]

"radial projection"
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}

\begin{align*}
\pi(x) \quad \text{Affine}_z \quad \| \quad \{ x : Ax = b \\
& \quad \text{and} \\
& \quad c^T x = z \} \end{align*}
\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]
\[ \min \ c^T x \]

s.t.
\[ Ax = b \]
\[ x \geq 0 \]

\[ \pi^* = \pi(x_z^*) \]

Affine\(_z\)
\[ \{ x : Ax = b \text{ and } c^T x = z \} \]
\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]
min $c^T x$

s.t. $Ax = b$

$x \geq 0$

$\pi(x) = 1 + \frac{1}{1-\min_j x_j}(x - 1)$
\[
\min \ c^T x \\
\text{s.t.} \ Ax = b \\
x \geq 0
\] \quad \text{LP}

\[
z^* \quad \text{optimal value, assumed finite}
\]

\[
z \quad \text{a fixed value satisfying } z < c^T 1
\]

\[
\text{Affine}_z \quad \{x : Ax = b \text{ and } c^T x = z\}
\]

\[
x \mapsto \pi(x) := 1 + \frac{1}{1 - \min_j x_j} (x - 1)
\]

\[
c^T \pi(x) = c^T 1 + \frac{1}{1 - \min_j x_j} (c^T x - c^T 1)
\]

\[
= c^T 1 + \frac{1}{1 - \min_j x_j} \left( z - c^T 1 \right)
\]

\[
a \text{negative constant}
\]

Thus, for \( x, y \in \text{Affine}_z \), \( c^T \pi(x) < c^T \pi(y) \Leftrightarrow \min_j x_j > \min_j y_j \)

\[
\textbf{Theorem:} \quad \text{LP is equivalent to}
\]

\[
\max_x \min_j x_j \quad \text{s.t.} \quad Ax = b \\
c^T x = z
\]

The only constraints in the equivalent problem are linear equations.
It’s thus easy to project onto the feasible region for the equivalent problem.
\[
\begin{align*}
\min_{x} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
\quad & \quad x \geq 0
\end{align*}
\]

\[
\max_{x} \quad \min_{j} x_{j}
\]

\[
\begin{align*}
\text{s.t.} & \quad Ax = b \\
& \quad c^T x = z
\end{align*}
\]

\[
x \mapsto \min_{j} x_{j} \quad \text{is the exemplary nonsmooth concave function}
\]

\begin{itemize}
\item Lipschitz continuous with constant \( M = 1 \)
\item Supgradients at \( x \) are the convex combinations of the standard basis vectors \( e(k) \) for which \( x_k = \min_{j} x_{j} \)
\end{itemize}

Thus, projected supgradients at \( x \) are the convex combinations of the corresponding columns of the projection matrix

\[
\bar{P} := I - \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A} \quad \text{where} \quad \bar{A} = \begin{bmatrix} A & c^T \end{bmatrix}
\]

Hence, in implementing a supgradient method, one option in choosing a supgradient at \( x \)
is simply to compute any column \( \bar{P}_k \) for which \( x_k = \min_{j} x_{j} \)

\[
x_+ = x + \frac{\epsilon}{\|\bar{P}_k\|^2} \bar{P}_k
\]

For large \( n \), compute columns of \( \bar{P} \) as needed do not (cannot) compute (store in memory) all of \( \bar{P} \)

With a modest amount of preprocessing work, the cost of each iteration is proportional to the number of nonzero entries in \( A \).
Now consider a semidefinite program, and for simplicity, assume the identity matrix is feasible.
\[
\begin{align*}
\min \quad & \langle C, X \rangle \\
\text{s.t.} \quad & \mathcal{A}(X) = b \\
& X \succeq 0
\end{align*}
\]
\[
\begin{align*}
\min \ & \langle C, X \rangle \\
\text{s.t.} \ & \mathcal{A}(X) = b \\
& X \succeq 0
\end{align*}
\]
equivalent problem

\[
\begin{align*}
\max \ & \lambda_{\min}(X) \\
\text{s.t.} \ & \mathcal{A}(X) = b \\
& \langle C, X \rangle = z
\end{align*}
\]

\[
\pi(X) = I + \frac{1}{1-\lambda_{\min}(X)}(X - I)
\]

\{
X : \mathcal{A}(X) = b
\text{ and }
\langle C, X \rangle = z
\}
Goal: Compute $X$ satisfying $\frac{\langle C, \pi(X) \rangle - z^*}{\langle C, I \rangle - z^*} \leq \epsilon$

To accomplish this, how accurately does $\lambda_{\min}(X)$ need to approximate $\lambda_{\min}(X^*_z)$?
\begin{align*}
\min & \quad \langle C, X \rangle \\
\text{s.t.} & \quad A(X) = b \\
X & \succeq 0
\end{align*} \quad \text{SDP}

\begin{align*}
\frac{\langle C, \pi(X) \rangle - z^*}{\langle C, I \rangle - z^*} \leq \epsilon & \iff \lambda_{\min}(X^*_z) - \lambda_{\min}(X) \leq \frac{\epsilon}{1 - \epsilon} \frac{\langle C, I \rangle - z}{\langle C, I \rangle - z^*}
\end{align*}

To get around the high accuracy required if the ratio is small, and to get around having to know the ratio (that is, having to know the optimal value $z^*$), we apply a supgradient method to multiple layers (details of which can be found in the arXiv posting) \ldots
initial user-supplied SDP-feasible matrix $X_0$

\[ \begin{align*}
\max & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*} \]
\[
\begin{align*}
\max & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*}
\]
\[
\begin{align*}
\max & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*}
\]
\[ \begin{align*} 
\max \ & \lambda_{\min}(X) \\
\text{s.t.} \ & A(X) = b \\
& \langle C, X \rangle = z 
\end{align*} \]
\[
\begin{aligned}
\text{max} & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{aligned}
\]
\[
\begin{align*}
\text{max } & \quad \lambda_{\text{min}}(X) \\
\text{s.t. } & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*}
\]
\[
\begin{align*}
\max & \quad \lambda_{\text{min}}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*}
\]
\[ \begin{align*}
\text{max} & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
& \quad \langle C, X \rangle = z
\end{align*} \]
\[ \begin{align*} \max & \quad \lambda_{\min}(X) \\ \text{s.t.} & \quad A(X) = b \\ & \quad \langle C, X \rangle = z \end{align*} \]
\[
\text{Diam} := \text{supremum of diameters of level sets for objective values } \leq \langle C, X_0 \rangle
\]
\[
\begin{align*}
\min & \quad \langle C, X \rangle \\
\text{s.t.} & \quad A(X) = b \\
X & \succeq 0
\end{align*}
\quad \text{SDP}
\]

\[
\langle C, \pi(X) \rangle - z^* \leq \epsilon \quad \Leftrightarrow \quad \frac{\lambda_{\min}(X^*_z) - \lambda_{\min}(X)}{\langle C, I \rangle - z^*} \leq \frac{\epsilon}{1 - \epsilon} \frac{\langle C, I \rangle - z}{\langle C, I \rangle - z^*}
\]

**Thm:**

\[
\ell \geq 8 \text{Diam}^2 \cdot \left( \frac{1}{\epsilon^2} + \frac{1}{\epsilon} \log_{4/3} \left( \frac{\langle C, I \rangle - z^*}{\langle C, I \rangle - \langle C, X_0 \rangle} \right) + 1 \right)
\]

\[
\Rightarrow \quad \min_{k \leq \ell} \frac{\langle C, \pi(X_k) \rangle - z^*}{\langle C, I \rangle - z^*} \leq \epsilon
\]
Now consider a general convex conic optimization problem, and fix a strictly feasible point $e$. 

$$\begin{align*}
\min & \quad c \cdot x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \mathcal{K} \quad \text{closed, convex cone with nonempty interior}
\end{align*}$$
Prop: The map \( x \mapsto \lambda_{\text{min}}(x) \) is concave and Lipschitz continuous.

\[
\min c \cdot x \\
\text{s.t. } Ax = b \\
x \in \mathcal{K}
\]

...where \( \lambda_{\text{min}}(x) \) is the scalar \( \lambda \) satisfying
\[
x - \lambda e \in \text{boundary}(\mathcal{K})
\]

\[
e + \frac{1}{1-\lambda_{\text{min}}(x)} (x - e)
\]
\[
\begin{array}{ll}
\text{min } & c \cdot x \\
\text{s.t. } & Ax = b \\
& x \in \mathcal{K} \\
\end{array}
\begin{array}{ll}
\text{max } & \lambda_{\min}(x) \\
\text{s.t. } & Ax = b \\
& c \cdot x = z
\end{array}
\]

\[
\frac{c \cdot \pi(x) - z^*}{c \cdot e - z^*} \leq \epsilon \iff \lambda_{\min}(x^*_z) - \lambda_{\min}(x) \leq \frac{\epsilon}{1 - \epsilon} \frac{c \cdot e - z}{c \cdot e - z^*}
\]

even in this general setting
we have “if and only if”

Whereas for linear programming we relied on the dot product,
and for SDP we relied on the trace product,
in this general setting
we allow computations to be done with respect to any inner product.

However, the extent to which the inner product
reflects the geometry of the cone $\mathcal{K}$ affects the Lipschitz constant \ldots
Prop: \[ |\lambda_{\min}(x) - \lambda_{\min}(y)| \leq \frac{1}{r_e} \|x - y\| \text{ for all } x, y \in \text{Affine}_z \]
and for every \( z \)

(see arXiv posting for full explanation)
initial user-supplied feasible point

$x_0$

e

$\text{Diam} := \sup \{ \text{diameters of level sets for objective values} \} \leq c \cdot x_0$

\[= \text{level sets}\]
\[
\begin{array}{ll}
\min & c \cdot x \\
\text{s.t.} & Ax = b \\
& x \in \mathcal{K} \\
\frac{c \cdot \pi(x) - z^*}{c \cdot e - z^*} & \leq \epsilon \\
\iff & \lambda_{\min}(x^*) - \lambda_{\min}(x) \leq \frac{\epsilon}{1 - \epsilon} \frac{c \cdot e - z}{c \cdot e - z^*} \\
\max & \lambda_{\min}(x) \\
\text{s.t.} & Ax = b \\
& c \cdot x = z
\end{array}
\]

Applying a supgradient method results in a sequence \(x_0, x_1, \ldots\) for which ...

**Thm:**

Lipschitz constant \(\leq 1/r_e\)

\[
\ell \geq 8 (M \text{ Diam})^2 \cdot \left( \frac{1}{\epsilon^2} + \frac{1}{\epsilon} \log_{4/3} \left( \frac{c \cdot e - z^*}{c \cdot e - c \cdot x_0} \right) + 1 \right)
\]

\[
\Rightarrow \min_{k \leq \ell} \frac{c \cdot \pi(x_k) - z^*}{c \cdot e - z^*} \leq \epsilon
\]
We depend heavily on various works of Nesterov, as well as results from the literature on “hyperbolic polynomials.”

Our motivation now is to develop an approach similar to the one of Nesterov, but which applies to optimization problems with complicated feasible regions rather than just “simple” ones.

We depend heavily on various works of Nesterov, as well as results from the literature on “hyperbolic polynomials.”

(Our most recent arXiv posting has all of the details.)
Smooth minimization of non-smooth functions

Received: February 4, 2003 / Accepted: July 8, 2004
Published online: December 29, 2004 – © Springer-Verlag 2004

Abstract. In this paper we propose a new approach for constructing efficient schemes for non-smooth convex optimization. It is based on a special smoothing technique, which can be applied to functions with explicit max-structure. Our approach can be considered as an alternative to black-box minimization. From the viewpoint of efficiency estimates, we manage to improve the traditional bounds on the number of iterations of the gradient schemes from $O\left(\frac{1}{\epsilon^2}\right)$ to $O\left(\frac{1}{\epsilon}\right)$, keeping basically the complexity of each iteration unchanged.

Yurii Nesterov

Smoothing technique and its applications in semidefinite optimization

Received: 20 January 2005 / Accepted: 23 February 2006 /
Published online: 27 April 2006
© Springer-Verlag 2006
Smoothing

Following Nesterov, rely on the smooth concave function

\[ f_\mu(X) := -\mu \ln \sum_j \exp(-\lambda_j(X)/\mu) \quad \text{(for fixed } \mu > 0) \]

Easy to see: \[ \lambda_{\min}(X) - \mu \ln n \leq f_\mu(X) \leq \lambda_{\min}(X) \]

Not so obvious, but which Nesterov showed:

\[ \| \nabla f_\mu(X) - \nabla f_\mu(Y) \|^* \leq \frac{1}{\mu} \|X - Y\|_\infty \]

that is, \( X \mapsto \nabla f_\mu(X) \) has Lipschitz constant \( L = 1/\mu \)

\[ \nabla f_\mu(X) = \sum_j \frac{1}{\exp(-\lambda_j(X)/\mu)} \begin{bmatrix} \exp(-\lambda_1(X)/\mu) \\ \vdots \\ \exp(-\lambda_n(X)/\mu) \end{bmatrix} Q^T \]

where \( X = Q \begin{bmatrix} \lambda_1(X) \\ \vdots \\ \lambda_n(X) \end{bmatrix} Q^T \) is an eigendecomposition of \( X \)

\textit{expensive!}

A relevant line of work thus begins with . . .

d’Aspremont, “Smooth optimization with approximate gradient”

Smoothing

Following Nesterov, rely on the smooth concave function

\[ f_\mu(X) := -\mu \ln \sum_j \exp(-\lambda_j(X)/\mu) \quad (\text{for fixed } \mu > 0) \]

Easy to see:

\[ \lambda_{\min}(X) - \mu \ln n \leq f_\mu(X) \leq \lambda_{\min}(X) \]

Not so obvious, but which Nesterov showed:

\[ \|\nabla f_\mu(X) - \nabla f_\mu(Y)\|^* \leq \frac{1}{\mu} \|X - Y\|_\infty \]

that is, \( X \mapsto \nabla f_\mu(X) \) has Lipschitz constant \( L = 1/\mu \)

\[ \nabla f_\mu(X) = \frac{1}{\sum_j \exp(-\lambda_j(X)/\mu)} Q \begin{bmatrix} \exp(-\lambda_1(X)/\mu) \\ \vdots \\ \exp(-\lambda_n(X)/\mu) \end{bmatrix} Q^T \]

where \( X = Q \begin{bmatrix} \lambda_1(X) \\ \vdots \\ \lambda_n(X) \end{bmatrix} Q^T \) is an eigendecomposition of \( X \)

For linear programming:

\[ \nabla f_\mu(x) = \frac{1}{\sum_j \exp(-x_j/\mu)} \begin{bmatrix} \exp(-x_1/\mu) \\ \vdots \\ \exp(-x_n/\mu) \end{bmatrix} \quad \text{inexpensive!} \]
\[
\begin{align*}
\min & \quad \langle C, X \rangle \\
\text{s.t.} & \quad A(X) = b \\
X \geq 0
\end{align*}
\equiv
\begin{align*}
\max & \quad \lambda_{\min}(X) \\
\text{s.t.} & \quad A(X) = b \\
\langle C, X \rangle = z
\end{align*}
\approx
\begin{align*}
\max & \quad f_{\mu}(X) \\
\text{s.t.} & \quad A(X) = b \\
\langle C, X \rangle = z
\end{align*}
\]

**Same goal as before:** Compute feasible \( X \) satisfying \( \frac{\langle C, X \rangle - z^*}{\langle C, I \rangle - z^*} \leq \epsilon \)

Choosing \( \mu = \epsilon/(6 \ln n) \)
and relying on Nesterov’s original accelerated gradient method . . .

**Thm:** A universal constant
\[
k \geq \kappa \cdot \left( \sqrt{\ln n} \cdot \text{Diam} \cdot \left( \frac{1}{\epsilon} + \log \frac{\langle C, I \rangle - z^*}{\langle C, I \rangle - \langle C, X_0 \rangle} \right) \right)
\]
\[
\Rightarrow \quad \frac{\langle C, \pi(X_k) \rangle - z^*}{\langle C, I \rangle - z^*} \leq \epsilon
\]

Especially-notable earlier work with similar iteration bounds:


Lan, Lu and Monteiro, “Primal-dual first-order methods with \( O(1/\epsilon) \) iteration-complexity for cone programming” *Math Prog* (2011)
\[
\min \ c \cdot x \\
\text{s.t.} \ Ax = b \\
x \in \mathcal{K}
\]

where \( \lambda_{\min}(x) \) is the scalar \( \lambda \) satisfying \( x - \lambda e \in \text{boundary}(\mathcal{K}) \)
\[
\begin{align*}
\text{min} & \quad c \cdot x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in K
\end{align*}
\]

Euclidean space

**Defn:** \( \mathcal{K} \subseteq \mathcal{E} \) is a “hyperbolicity cone” if there is a homogeneous polynomial \( p \) satisfying:

- \( p(x) \neq 0 \) for all \( x \in \text{int}(\mathcal{K}) \)
- \( p(x) = 0 \) for all \( x \in \text{bdy}(\mathcal{K}) \)
- \( \exists e \in \text{int}(\mathcal{K}) \) such that for all \( x \in \mathcal{E} \), the univariate polynomial \( \lambda \mapsto p(x - \lambda e) \) has only real roots.

**Example:** \( \mathcal{K} = S_{+}^{n \times n} \), \( p(X) = \det(X) \), \( e = I \), \( \lambda \mapsto \det(X - \lambda I) \)

Also: non-negative orthant, second-order cones, and many others.

**Theorem (Gårding, 1959):** For every \( e \in \text{int}(\mathcal{K}) \) and all \( x \in \mathcal{E} \), the univariate polynomial \( \lambda \mapsto p(x - \lambda e) \) has only real roots.

If each of \( \mathcal{K}_1 \) and \( \mathcal{K}_2 \) is a hyperbolicity cone, then so is \( \mathcal{K}_1 \times \mathcal{K}_2 \) and\( ^* \mathcal{K}_1 \cap \mathcal{K}_2 \).

If \( \mathcal{K}' \subseteq \mathcal{E}' \) is a hyperbolicity cone and \( T : \mathcal{E} \to \mathcal{E}' \) is a linear transformation, then\( ^* \mathcal{K} := \{x : T(x) \in \mathcal{K}'\} \) is a hyperbolicity cone.
“hyperbolic program”

\[
\begin{align*}
\text{min} & \quad c \cdot x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \mathcal{K} \quad \text{hyperbolicity cone}
\end{align*}
\]

\[
\min c \cdot x \\
\text{s.t. } Ax = b \\
\quad x \in \mathcal{K}
\]

Now every \( x \) has \( n \) real \text{“eigenvalues”} \( \lambda_1(x), \ldots, \lambda_n(x) \),

\[e + \frac{1}{1-\lambda_{\min}(x)} (x - e)\]
\[ f_\mu(x) := -\mu \ln \sum_j \exp(-\lambda_j(x)/\mu) \quad \text{(for fixed } \mu > 0) \]

Easy to see:
\[ \lambda_{\min}(x) - \mu \ln n \leq f_\mu(x) \leq \lambda_{\min}(x) \]

**Prop:** \( f_\mu \) is concave and infinitely Fréchet differentiable.

Moreover, \[ \| \nabla f_\mu(x) - \nabla f_\mu(y) \|^* \leq \frac{1}{\mu} \| x - y \|_\infty \quad \text{for all } x, y. \]

(Mostly a corollary to Nesterov’s result and the Helton-Vinnikov Theorem.)

Moreover, the gradients are readily computable, especially if the underlying “hyperbolic polynomial” can be factored as the product of polynomials of low degrees.

(see the arXiv posting for details)
Cor: \[ \| P \nabla f_\mu(x) - P \nabla f_\mu(y) \| \leq \frac{1}{r_e^2 \mu} \| x - y \| \quad \text{for all } x, y \in \text{Affine}_{e} \]
and for every \( z \).
where $L$ is the Lipschitz constant for the gradient map and where $L_0 > 0$ is the input guess of $L$.

Same goal as before: Compute feasible $x$ satisfying

$$
\min c \cdot x \\
\text{s.t. } Ax = b \\
\phantom{\text{s.t. }} x \in \mathcal{K}
$$

$\equiv$

$$
\max \lambda_{\min}(x) \\
\text{s.t. } Ax = b \\
\phantom{\text{s.t. }} c \cdot x = z \\
\approx$

$$
\max f_\mu(x) \\
\text{s.t. } Ax = b \\
\phantom{\text{s.t. }} c \cdot x = z
$$

Choosing $\mu = \epsilon / (6 \ln n)$ and using a “uniformly optimal” (or “universal”) accelerated gradient method (Lan (2010), Nesterov (2014)) ... 

Thm:

The algorithm produces $x_k$ satisfying

$$
\frac{c \cdot x_k - z^*}{c \cdot e - z^*} \leq \epsilon
$$

and does so within computing a total number of gradients not exceeding

$$
\mathcal{O}\left(\text{Diam} \cdot \sqrt{L} \cdot \left(\frac{1}{\sqrt{\epsilon}} + \log \frac{c \cdot e - z^*}{c \cdot e - c \cdot x_0} + \left| \log \frac{L}{L'} \right| \right)\right)
$$

where $L$ is the Lipschitz constant for the gradient map and where $L' > 0$ is the input guess of $L$.

Note: $L \leq \frac{1}{r^2 \mu} = \frac{6 \ln n}{r^2 \epsilon}$
Let’s see what results by applying the framework to general convex optimization problems by putting those problems into conic form.

First we consider minimizing a convex function subject to no constraints, but we make some assumptions on the function so that we can clarify how the new approach differs from applying subgradient methods directly . . .
Assume:

- $f$ is lower semicontinuous and has a minimizer

$$\min f(x)$$
\[ \min f(x) \equiv \min_{x,t} t \]
\[ \text{s.t. } (x,t) \in \text{epi}(f) := \{(x,t) : f(x) \leq t\} \]
\[ \text{closed convex set} \]

\[ \min_{x,t,t'} t \]
\[ \text{s.t. } t' = 1 \]
\[ (x,t,t') \in \mathcal{K} \]

where \( \mathcal{K} \) is the closed cone for which
\[ (x,t,1) \in \mathcal{K} \iff (x,t) \in \text{epi}(f) \]

The third problem is a conic formulation
to which we apply our supgradient framework
\[ \min f(x) \]

Assume:
- \( f \) is lower semicontinuous and has a minimizer
- \( \{x : f(x) < \infty\} \) is open
- \( \|x\| < 1 \Rightarrow f(x) < 0 \)

Graph of such a function \( f \)
\[ \min f(x) \]

Assume:
- \( f \) is lower semicontinuous and has a minimizer
- \( \{ x : f(x) < \infty \} \) is open
- \( \| x \| < 1 \Rightarrow f(x) < 0 \)

Initialize: \( x_0 = \vec{0}, \ z = f(\vec{0}) \)

Iterate:
1. Compute the positive scalar \( \alpha_k \) satisfying \( f(\alpha_k x_k) = \alpha_k z \).
2. Let \( y_k := \alpha_k x_k \).
$z < 0$ – an upper bound on the optimal value of $\min f(x)$

(the value $z$ is occasionally updated)

Determine the positive scalar $\alpha_k$ for which $f(\alpha_k x_k) = \alpha_k z$,
and then define $y_k = \alpha_k x_k$. 

$\mathbb{R}^n$

\[ \vec{0} \quad y_k := \frac{1}{2} x_k \quad x_k \]
$z < 0$ – an upper bound on the optimal value of $\min f(x)$

(the value $z$ is occasionally updated)

If $\alpha_k < 4/3$, then let $x_{k+1} = x_k + \frac{\epsilon}{2\|g\|^2} g$

where $g = \frac{1}{f(y_k) + \langle \nabla f(y_k) , \bar{0} - y_k \rangle} \nabla f(y_k)$

The subgradient is for $y_k$ but the step is taken from $x_k$!
$z < 0$ – an upper bound on the optimal value of $\min f(x)$ (the value $z$ is occasionally updated)

Determine the positive scalar $\alpha_{k+1}$ for which $f(\alpha_{k+1}x_{k+1}) = \alpha_{k+1}z$, and then define $y_{k+1} = \alpha_{k+1}x_{k+1}$.
$z < 0$ – an upper bound on the optimal value of $\min f(x)$

(the value $z$ is occasionally updated)

If $\alpha_{k+1} \geq 4/3$, define $x_{k+2} = y_{k+1}$ and update $z$: $z \leftarrow f(x_{k+2})$
Assume:

- \( f \) is lower semicontinuous and has a minimizer
- \( \{ x : f(x) < \infty \} \) is open
- \( \|x\| < 1 \Rightarrow f(x) < 0 \)

Initialize: \( x_0 = \bar{0}, \ z = f(\bar{0}) \)

Iterate:

1. Compute the positive scalar \( \alpha_k \) satisfying \( f(\alpha_k x_k) = \alpha_k z \).

2. Let \( y_k := \alpha_k x_k \).

3. If \( \alpha_k \geq 4/3 \), let \( x_{k+1} = y_k \) and \( z \leftarrow f(y_k) \).

Else let \( x_{k+1} = x_k + \frac{\epsilon}{2\|g_k\|^2} g_k \)

where \( g_k = \frac{1}{\nabla f(y_k) + (\nabla f(y_k), \bar{0} - y_k)} \nabla f(y_k) \).

\( \nabla f(y_k) \leq f(\bar{0}) < 0 \) subgradient at \( y_k \)
Assume:

\[
\min f(x) \\
\begin{align*}
\bullet & \ f \text{ is lower semicontinuous and has a minimizer} \\
\bullet & \ \{x : f(x) < \infty\} \text{ is open} \\
\bullet & \ \|x\| < 1 \Rightarrow f(x) < 0
\end{align*}
\]

Cor:

The supgradient algorithm computes \( y_k \) satisfying
\[
\frac{f(y_k) - f^*}{0 - f^*} \leq \epsilon
\]

where \( k \leq 8D^2 \left( \frac{1}{\epsilon^2} + \frac{1}{\epsilon} \log_{4/3} \left( (D + 1) (1 - \epsilon) \right) + 1 \right) \)

\[
\text{defining } D = \text{diameter} \left( \{x : f(x) \leq f(\tilde{0})\} \right).
\]

\textit{Differs from traditional subgradient literature in that } f \textit{ is not required to be Lipschitz continuous.}
More generally . . . extended valued, convex, lower semicontinuous

\[
\begin{align*}
\min \ f(x) \\
\text{s.t. } \ x \in \text{Feas} 
\end{align*}
\]
\[\{x \in S : Ax = b\}\]
\[\text{closed and convex}\]

Assume:

- $\bar{x}$ satisfies $A\bar{x} = b$ and $\bar{x} \in \text{interior}(S \cap \text{effective}\_\text{domain}(f))$
- Euclidean norm satisfies $\{x \in B(\bar{x}, 1) : Ax = b\} \subseteq S \cap \text{effective}\_\text{domain}(f)$
  - let $\hat{f}$ be a scalar upper bound on $f(x)$ for all $x$ in this set
- $D = \text{diameter}(\{x \in \text{Feas} : f(x) \leq f(\bar{x})\})$

Then can compute feasible $x$ satisfying $\frac{f(x) - f^*}{\hat{f} - f^*} \leq \epsilon$

within $O\left(D^2 \cdot \left(\frac{1}{\epsilon^2} + \frac{1}{\epsilon} \log D\right)\right)$ iterations.

(see arXiv posting for details)
But the most important takeaway is entirely elementary ...
\[
\min \quad c^T x \\
\text{s.t.} \quad Ax = b \\
x \in K
\]

...where \( \lambda_{\text{min}}(x) \) is the scalar \( \lambda \) satisfying \( x - \lambda e \in \text{boundary}(K) \)

\[
\min \quad c \cdot x \\
\text{s.t.} \quad Ax = b \\
x \in K
\]
\[
\equiv \\
\max \quad \lambda_{\text{min}}(x) \\
\text{s.t.} \quad Ax = b \\
c \cdot x = z
\]

Thanks for listening!