Localization and Cutting-plane Methods

- idea of localization methods
- bisection on $\mathbb{R}$
- center of gravity algorithm
- analytic center cutting-plane method

Prof. S. Boyd, EE392o, Stanford University
Localization

- \( f : \mathbb{R}^n \to \mathbb{R} \) convex (and for now, differentiable)
- **problem:** minimize \( f \)
- **oracle model:** for any \( x \) we can evaluate \( f \) and \( \nabla f(x) \) (at some cost)

from \( f(x) \geq f(x_0) + \nabla f(x_0)^T (x - x_0) \) we conclude

\[
\nabla f(x_0)^T (x - x_0) \geq 0 \quad \implies \quad f(x) \geq f(x_0)
\]

i.e., all points in halfspace \( \nabla f(x_0)^T (x - x_0) \geq 0 \) are **worse** than \( x_0 \)
level curves of $f$

\[ \nabla f(x_0) \]

\[ \nabla f(x_0)^T(x - x_0) \geq 0 \]

- by evaluating $\nabla f$ we rule out a halfspace in our search for $x^*$:

\[ x^* \in \{ x \mid \nabla f(x_0)^T(x - x_0) \leq 0 \} \]

- **idea:** get one bit of info (on location of $x^*$) by evaluating $\nabla f$

- for nondifferentiable $f$, can replace $\nabla f(x_0)$ with any subgradient $g \in \partial f(x_0)$
suppose we have evaluated $\nabla f(x_1), \ldots, \nabla f(x_k)$
then we know $x^* \in \{x \mid \nabla f(x_i)^T(x - x_i) \leq 0\}$
on the basis of $\nabla f(x_1), \ldots, \nabla f(x_k)$, we have localized $x^*$ to a polyhedron
**question:** what is a ‘good’ point $x_{k+1}$ at which to evaluate $\nabla f$?
Localization algorithm

basic (conceptual) localization (or cutting-plane) algorithm:

1. after iteration $k - 1$ we know $x^* \in \mathcal{P}_{k-1}$:

$$\mathcal{P}_{k-1} = \{ x \mid \nabla f(x^{(i)})^T (x - x^{(i)}) \leq 0, \ i = 1, \ldots, k - 1 \}$$

2. evaluate $\nabla f(x^{(k)})$ (or $g \in \partial f(x^{(k)})$) for some $x^{(k)} \in \mathcal{P}_{k-1}$

3. $\mathcal{P}_k := \mathcal{P}_{k-1} \cap \{ x \mid \nabla f(x^{(k)})^T (x - x^{(k)}) \leq 0 \}$
$P_{k-1}$

$\nabla f(x^{(k)})$

$x^{(k)}$

$P_k$

$\nabla f(x^{(k)})$

$x^{(k)}$

$P_k$ gives our uncertainty of $x^*$ at iteration $k$

want to pick $x^{(k)}$ so that $P_{k+1}$ is as small as possible

clearly want $x^{(k)}$ near center of $C^{(k)}$
Example: bisection on $\mathbb{R}$

- $f : \mathbb{R} \rightarrow \mathbb{R}$
- $\mathcal{P}_k$ is interval
- obvious choice: $x^{(k+1)} := \text{midpoint}(\mathcal{P}_k)$

\begin{center}
\begin{tcolorbox}
\textbf{bisection algorithm}
\\
given interval $C = [l, u]$ containing $x^*$
repeat
\begin{enumerate}
  \item $x := (l + u)/2$
  \item evaluate $f'(x)$
  \item if $f'(x) < 0$, $l := x$; else $u := x$
\end{enumerate}
\end{tcolorbox}
\end{center}
\[ x^{(k+1)} \]
\[ \text{length}(\mathcal{P}_{k+1}) = u_{k+1} - l_{k+1} = \frac{u_k - l_k}{2} = (1/2)\text{length}(\mathcal{P}_k) \]

and so \(\text{length}(\mathcal{P}_k) = 2^{-k}\text{length}(\mathcal{P}_0)\)

**interpretation:**

- \(\text{length}(\mathcal{P}_k)\) measures our uncertainty in \(x^*\)
- uncertainty is halved at each iteration; get exactly one bit of info about \(x^*\) per iteration
- \# steps required for uncertainty (in \(x^*\)) \(\leq \epsilon\):
  \[
  \log_2 \frac{\text{length}(\mathcal{P}_0)}{\epsilon} = \log_2 \frac{\text{initial uncertainty}}{\text{final uncertainty}}
  \]
question:

• can bisection be extended to $\mathbb{R}^n$?

• or is it special since $\mathbb{R}$ is linear ordering?
Center of gravity algorithm

take $x^{(k+1)} = \text{CG}(P_k)$ (center of gravity)

$$\text{CG}(P_k) = \frac{\int_{P_k} x \, dx}{\int_{P_k} dx}$$

**Theorem.** if $C \subseteq \mathbb{R}^n$ convex, $x_{cg} = \text{CG}(C)$, $g \neq 0$,

$$\text{vol}\left(C \cap \{x \mid g^T(x - x_{cg}) \leq 0\}\right) \leq (1 - 1/e) \text{vol}(C') \approx 0.63 \, \text{vol}(C')$$

(independent of dimension $n$)

hence in CG algorithm, $\text{vol}(P_k) \leq 0.63^k \, \text{vol}(P_0)$
• \( \text{vol}(\mathcal{P}_k)^{1/n} \) measures uncertainty (in \( x^* \)) at iteration \( k \)

• uncertainty reduced at least by \( 0.63^{1/n} \) each iteration

• from this can prove \( f(x^{(k)}) \rightarrow f(x^*) \) (later)

• max. \# steps required for uncertainty \( \leq \epsilon \):

\[
1.51n \log_2 \frac{\text{initial uncertainty}}{\text{final uncertainty}}
\]

(cf. bisection on \( \mathbb{R} \))
advantages of CG-method

- guaranteed convergence

- number of steps proportional to dimension $n$, log of uncertainty reduction

disadvantages

- finding $x^{(k+1)} = \text{CG}(\mathcal{P}_k)$ is harder than original problem

- $\mathcal{P}_k$ becomes more complex as $k$ increases
  (removing redundant constraints is harder than solving original problem)

(but, can modify CG-method to work)
Analytic center cutting-plane method

**analytic center** of polyhedron $\mathcal{P} = \{ z \mid a_i^T z \preceq b_i, \ i = 1, \ldots, m \}$ is

$$AC(\mathcal{P}) = \arg\min_z - \sum_{i=1}^m \log(b_i - a_i^T z)$$

**ACCPM** is localization method with next query point $x^{(k+1)} = AC(\mathcal{P}_k)$ (found by Newton’s method)
Outer ellipsoid from analytic center

• let $x^*$ be analytic center of $\mathcal{P} = \{z \mid a_i^T z \preceq b_i, \ i = 1, \ldots, m\}$

• let $H^*$ be Hessian of barrier at $x^*$,

$$H^* = -\nabla^2 \sum_{i=1}^m \log(b_i - a_i^T z) \bigg|_{z=x^*} = \sum_{i=1}^m \frac{a_i a_i^T}{(b_i - a_i^T x^*)^2}$$

• then, $\mathcal{P} \subseteq \mathcal{E} = \{z \mid (z - x^*)^T H^*(z - x^*) \leq m^2\}$ (not hard to show)
Lower bound in ACCPM

let $\mathcal{E}^{(k)}$ be outer ellipsoid associated with $x^{(k)}$

a lower bound on optimal value $p^*$ is

$$p^* \geq \inf_{z \in \mathcal{E}^{(k)}} \left( f(x^{(k)}) + g^{(k)T}(z - x^{(k)}) \right)$$

$$= f(x^{(k)}) - m_k \sqrt{g^{(k)T} H^{(k)} - 1} g^{(k)}$$

($m_k$ is number of inequalities in $\mathcal{P}_k$)

gives simple stopping criterion $\sqrt{g^{(k)T} H^{(k)} - 1} g^{(k)} \leq \epsilon / m_k$
**Best objective and lower bound**

since ACCPM isn't a descent a method, we keep track of best point found, and best lower bound

best function value so far: $u_k = \min_{i=1,\ldots,k} f(x^{(k)})$

best lower bound so far: $l_k = \max_{i=1,\ldots,k} f(x^{(k)}) - m_k \sqrt{g^{(k)T} H^{(k)} - 1} g^{(k)}$

can stop when $u_k - l_k \leq \epsilon$
Basic ACCPM

given polyhedron $\mathcal{P}$ containing $x^*$

repeat
1. compute $x^*$, the analytic center of $\mathcal{P}$, and $H^*$
2. compute $f(x^*)$ and $g \in \partial f(x^*)$
3. $u := \min\{u, f(x^*)\}$
   \[ l := \max\{l, f(x^*) - m \sqrt{g^T H^*^{-1} g}\} \]
4. add inequality $g^T(z - x^*) \leq 0$ to $\mathcal{P}$

until $u - l < \epsilon$

here $m$ is number of inequalities in $\mathcal{P}$
Dropping constraints

add an inequality to $\mathcal{P}$ each iteration, so centering gets harder, more storage as algorithm progresses

schemes for dropping constraints from $\mathcal{P}^{(k)}$:

- remove all redundant constraints (expensive)
- remove some constraints known to be redundant
- remove constraints based on some relevance ranking
Dropping constraints in ACCPM

\( x^* \) is AC of \( \mathcal{P} = \{ x \mid a_i^T x \leq b_i, \; i = 1, \ldots, m \} \), \( H^* \) is barrier Hessian at \( x^* \)

Define (ir)relevance measure \( \eta_i = \frac{b_i - a_i^T x^*}{\sqrt{a_i^T H^* a_i}} \)

- \( \eta_i/m \) is normalized distance from hyperplane \( a_i^T x = b_i \) to outer ellipsoid
- If \( \eta_i \geq m \), then constraint \( a_i^T x \leq b_i \) is redundant
common ACCPM constraint dropping schemes:

- drop all constraints with $\eta_i \geq m$ (guaranteed to not change $\mathcal{P}$)

- drop constraints in order of irrelevance, keeping constant number, usually $3n - 5n$
Example

PWL objective, $n = 10$ variables, $m = 100$ terms
simple ACCPM: $f(x^{(k)})$ and lower bound $f(x^{(k)}) - m \sqrt{g^{(k)T}H^{(k)} - 1} g^{(k)}$
simple ACCPM: $u_k$ (best objective value) and $l_k$ (best lower bound)
ACCPM with constraint dropping

\[ u_k - p^* \]
\[ u_k - l_k \]

\[ k \]

... constraint dropping actually \textbf{improves} convergence (!)
ACCPM with constraint dropping

number of inequalities in $\mathcal{P}$:

Prof. S. Boyd, EE392o, Stanford University
Handling inequality constraints

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m
\end{align*}
\]

same idea: maintain polyhedron \( P^{(k)} \) that contains \( x^* \)

at each \( x \), need oracle to give cutting-plane that separates \( x \) from \( x^* \), i.e., \( g \neq 0 \) with

\[
g^T(x^* - x) \leq 0
\]
Cutting-plane oracle for problem with inequalities

case 1: \(x^{(k)}\) feasible, i.e., \(f_i(x^{(k)}) \leq 0, i = 1, \ldots, m\)

- take cutting plane \(g = \nabla f_0(x^{(k)})\) (or \(g \in \partial f_0(x^{(k)})\))
- rules out halfspace of points with larger function value than current point

\[g \leq 0 \quad \Rightarrow \quad x \text{ infeasible}\]

\[\Rightarrow \quad f_j(x) > 0 \quad \Rightarrow \quad x \text{ infeasible, so take } g = \nabla f_j(x^{(k)}) \quad (\text{or } g \in \partial f_j(x^{(k)}))\]

\[g \leq 0 \quad \Rightarrow \quad x \text{ infeasible}\]

- rules out halfspace of infeasible points

case 2: \(x^{(k)}\) infeasible, say, \(f_j(x^{(k)}) > 0;\)

- then \(\nabla f_j(x^{(k)})^T(x - x^{(k)}) \geq 0 \quad \Rightarrow \quad f_j(x) > 0 \quad \Rightarrow \quad x \text{ infeasible, so take } g = \nabla f_j(x^{(k)}) \quad (\text{or } g \in \partial f_j(x^{(k)}))\)
## Stopping criterion

if $x^{(k)}$ is feasible, we have a lower bound on $p^*$ as before:

$$
p^* \geq f_0(x^{(k)}) - m_k \sqrt{\nabla f_0(x^{(k)})^T H^{(k)} - 1 \nabla f_0(x^{(k)})}
$$

if $x^{(k)}$ is infeasible, we have for all $x \in \mathcal{E}^{(k)}$ (outer ellipsoid)

$$
f_j(x) \geq f_j(x^{(k)}) + \nabla f_j(x^{(k)})^T (x - x^{(k)})
$$

$$
\geq f_j(x^{(k)}) + \inf_{x \in \mathcal{E}^{(k)}} \nabla f_j(x^{(k)})^T (x - x^{(k)})
$$

$$
= f_j(x^{(k)}) - m_k \sqrt{\nabla f_j(x^{(k)})^T H^{(k)} - 1 \nabla f_j(x^{(k)})}
$$
hence, problem is infeasible if for some $j$,

$$f_j(x^{(k)}) - m_k \sqrt{\nabla f_j(x^{(k)})^T H^{(k)} - 1 \nabla f_j(x^{(k)})} > 0$$

**stopping criteria:**

- if $x^{(k)}$ is feasible and $m_k \sqrt{\nabla f_0(x^{(k)})^T H^{(k)} - 1 \nabla f_0(x^{(k)})} \leq \epsilon$ ($x^{(k)}$ is $\epsilon$-suboptimal)

- if $f_j(x^{(k)}) - m_k \sqrt{\nabla f_j(x^{(k)})^T H^{(k)} - 1 \nabla f_j(x^{(k)})} > 0$ (problem is infeasible)