layout: true class: typo, typo-selection

---

count: false class: nord-dark, middle, center

Ellipsoid Method Revisited

Wai-Shing Luk

2022-11-03

Some History of Ellipsoid Method [@BGT81]

- Introduced by Shor and Yudin and Nemirovskii in 1976
- Used to show that linear programming (LP) is polynomial-time solvable (Kachiyan 1979), settled the long-standing problem of determining the theoretical complexity of LP.
- In practice, however, the simplex method runs much faster than the method, although its worst-case complexity is exponential.

Basic Ellipsoid Method

- An ellipsoid $\mathcal{E}(x_c, P)$ is specified as a set

  $$x \mid (x - x_c)P^{-1}(x - x_c) \leq 1,$$

  where x_c is the center of the ellipsoid.

Updating the ellipsoid (deep-cut)

Calculation of minimum volume ellipsoid $\mathcal{E}^+$ covering:

$$\mathcal{E} \cap z \mid g^\mathsf{T}(z - x_c) + \beta \leq 0.$$

- Let $\tilde{g} = P\,g$, $\tau^2 = g^\mathsf{T}Pg$.
- If $n \cdot \beta < -\tau$ (shallow cut), no smaller ellipsoid can be found.
- If $\beta > \tau$, intersection is empty.

Otherwise,

$$x_c^+ = x_c - \frac{\rho}{\tau^2}\tilde{g}, \quad P^+ = \delta \cdot \left(P - \frac{\sigma}{\tau^2}\tilde{g}\tilde{g}^\mathsf{T}\right), \quad (P')^{-1} = \delta^{-1} \cdot \left(P^{-1} + \frac{\mu}{\tau^2}gg^\mathsf{T}\right).$$

where

$$\rho = \frac{\tau + n \cdot \beta}{n+1}, \quad \sigma = \frac{2\rho}{\tau + \beta}, \quad \delta = \frac{n^2(\tau + \beta)(\tau - \beta)}{(n^2 - 1)\tau^2}, \quad \mu = \frac{2(\tau + n \cdot \beta)}{(n-1)(\tau - \beta)}$$

Updating the ellipsoid (cont'd)

- Even better, split P into two variables $\kappa \cdot Q$
- Let $\tilde{g} = Q \cdot g$, $\omega = g^\mathsf{T}\tilde{g}$, $\tau = \sqrt{\kappa \cdot \omega}$.

$$x_c^+ = x_c - \frac{\rho}{\omega}\tilde{g}, \quad Q' = Q - \frac{\sigma}{\omega}\tilde{g}\tilde{g}^\mathsf{T}, \quad (Q')^{-1} = Q^{-1} + \frac{\mu}{\omega}gg^\mathsf{T}, \quad \kappa^+ = \delta \cdot \kappa.$$

- Reduce n^2 multiplications per iteration.
- Note:
  - The determinant of Q decreases monotonically.
  - The range of δ is $(0, \frac{n^2}{n^2-1})$.

Central Cut

- A Special case of deep cut when $\beta = 0$
- Deserve a separate implement because it is much simplier.
- Let $\tilde{g} = Q\,g$, $\tau = \sqrt{\kappa \cdot \omega}$,

$$\rho = \frac{\tau}{n+1}, \quad \sigma = \frac{2}{n+1}, \quad \delta = \frac{n^2}{n^2 - 1}, \quad \mu = \frac{2}{n-1}.$$

---

class: middle, center

Parallel Cuts

Parallel Cuts

- Oracle returns a pair of cuts instead of just one.
- The pair of cuts is given by g and (β_0, β_1) such that:

$$g^\mathsf{T}(x - x_c) + \beta_0 \leq 0,$$

$$g^\mathsf{T}(x - x_c) + \beta_1 \geq 0,$$

  for all $x \in \mathcal{K}$. $$

- Only linear inequality constraint can produce such parallel cut:

$$l \leq a^\mathsf{T} x + b \leq u, \quad L \preceq F(x) \preceq U.$$

- Usually provide faster convergence.

Updating the ellipsoid

- Let $\tilde{g} = Q\,g$, $\tau^2 = \kappa \cdot \omega$.
- If $\beta_0 > \beta_1$, intersection is empty.
- If $\beta_0 \beta_1 < -\tau^2/n$, no smaller ellipsoid can be found.
- If $\beta_1^2 > \tau^2$, it reduces to deep-cut with $\alpha = \alpha_1$
- Otherwise,

$$x_c' = x_c - \frac{\rho}{\omega}\tilde{g}, \quad Q' = Q - \frac{\sigma}{\omega}\tilde{g}\tilde{g}^\mathsf{T}, \quad (Q')^{-1} = Q^{-1} + \frac{\mu}{\omega} g g^\mathsf{T}, \quad \kappa^+ = \delta\kappa.$$

  where

$$\begin{array}{rcl}
\bar{\beta} &=& (\beta_0 + \beta_1)/2, \\
\xi^2 &=& (\tau^2 - \beta_0^2)(\tau^2 - \beta_1^2) + (n(\beta_1 - \beta_0)\bar{\beta})^2, \\
\sigma &=& (n + (\tau^2 + \beta_0\beta_1 - \xi)/(2\bar{\beta}^2))/(n+1), \\
\rho &=& \bar{\beta} \cdot \sigma, \\
\mu &=& \sigma/(1 - \sigma), \\
\delta &=& (n^2/(n^2 - 1))(\tau^2 - (\beta_0^2 + \beta_1^2)/2 + \xi/n)/\tau^2.
\end{array}$$

Example - FIR filter design

- The time response is:

$$y[t] = \sum_{k=0}^{n-1} h[k] u[t-k].$$

Example - FIR filter design (cont'd)

- The frequency response:

$$H(\omega) \;=\; \sum_{m=0}^{n-1} h(m)e^{-jm\omega}.$$

- The magnitude constraints on frequency domain are expressed as

$$L(\omega) \;\leq\; |H(\omega)| \;\leq\; U(\omega),\; \forall\, \omega \in (-\infty, +\infty).$$

  where $L(\omega)$ and $U(\omega)$ are the lower and upper (nonnegative) bounds at frequency ω respectively.

- The constraint is non-convex in general.

Example - FIR filter design (II)

- However, via *spectral factorization* [@goodman1997spectral], it can transform into a convex one [@wu1999fir]:

$$L^2(\omega) \;\leq\; R(\omega) \;\leq\; U^2(\omega),\; \forall\, \omega \in (0, \pi),$$

  where

  - $R(\omega) = \sum_{i=-1+n}^{n-1} r(t)e^{-j\omega t} = |H(\omega)|^2$
  - $\mathbf{r} = (r(-n+1), r(-n+2), ..., r(n-1))$ are the autocorrelation coefficients.

Example - FIR filter design (III)

- $\mathbf{r}$ can be determined by $\mathbf{h}$:

$$r(t) \;=\; \sum_{i=-n+1}^{n-1} h(i)h(i+t),\; t \in \mathbf{Z},$$

  where $h(t) = 0$ for $t < 0$ or $t > n-1$.

- The whole problem can be formulated as:

$$\begin{aligned} \min \quad & \gamma \\ \text{s.t.} \quad & L^2(\omega) \le R(\omega) \le U^2(\omega),\ \forall \omega \in [0, \pi] \\ & R(\omega) > 0, \forall \omega \in [0, \pi] \end{aligned}$$

Google Benchmark Result

```
3: ------------------------------------------------------------------
3: Benchmark                               Time             CPU   Iterations
3: ------------------------------------------------------------------
3: BM_Lowpass_single_cut     627743505 ns    621639313 ns            1
3: BM_Lowpass_parallel_cut    30497546 ns     30469134 ns           24
3/4 Test #3: Bench_BM_lowpass .................   Passed    1.72 sec
```

Example - Maximum Likelihood estimation

$$\begin{aligned} \min_{\kappa, p} \quad & \log\det(\Omega(p) + \kappa \cdot I) + \mathrm{Tr}((\Omega(p) + \kappa \cdot I)^{-1} Y) \\ \text{s.t.} \quad & \Omega(p) \succeq 0, \kappa \ge 0 \end{aligned}$$

Note: the 1st term is concave, the 2nd term is convex

- However, if there are enough samples such that Y is a positive definite matrix, then the function is convex within $[0, 2Y]$

Example - Maximum Likelihood estimation (cont'd)

- Therefore, the following problem is convex:

$$\begin{aligned} \min_{\kappa, p} \quad & \log\det V(p) + \mathrm{Tr}(V(p)^{-1} Y) \\ \text{s.t.} \quad & \Omega(p) + \kappa \cdot I = V(p) \\ & 0 \preceq V(p) \preceq 2Y, \kappa > 0 \end{aligned}$$

---

class: middle, center

Discrete Optimization

Why Discrete Convex Programming

- Many engineering problems can be formulated as a convex/geometric programming, e.g. digital circuit sizing
- Yet in an ASIC design, often there is only a limited set of choices from the cell library. In other words, some design variables are discrete.
- The discrete version can be formulated as a *Mixed-Integer Convex programming* (MICP) by mapping the design variables to integers.

What's Wrong w/ Existing Methods?

- Mostly based on relaxation.
- Then use the relaxed solution as a lower bound and use the branch–and–bound method for the discrete optimal solution.
  - Note: the branch-and-bound method does not utilize the convexity of the problem.
- What if I can only evaluate constraints on discrete data? Workaround: convex fitting?

Mixed-Integer Convex Programming

Consider:

$$\begin{array}{ll} \text{minimize} & f_0(x), \\ \text{subject to} & f_j(x) \leq 0,\ \forall j = 1, 2, \ldots \\ & x \in \mathbb{D} \end{array}$$

where

- $f_0(x)$ and $f_j(x)$ are "convex"
- Some design variables are discrete.

Oracle Requirement

- The oracle looks for the nearby discrete solution x_d of x_c with the cutting-plane:
$$g^\mathsf{T}(x - x_d) + \beta \leq 0, \beta \geq 0, g \neq 0$$
- Note: the cut may be a shallow cut.
- Suggestion: use different cuts as possible for each iteration (e.g. round-robin the evaluation of constraints)

Discrete Cut

Example - Multiplier-less FIR filter design (nnz=3)

---

class: nord-dark, middle, center

Q & A