- Stochastic programming

$$\min.\ E f_0(x,w)$$
$$\text{s.t.}\ E f_i(x,w) \le 0$$

a convex problem if f_i is convex in x for each w — expectation is integral, integral preserves convexity

$\Updownarrow$ "certainty-equivalent" problem

$$\min.\ f_0(x, E[w])$$
$$\text{s.t.}\ f_i(x, E[w]) \le 0$$

this is what's normally done, though not explicitly

gives a lower bound by Jensen's inequality

when the "certainty-equivalent" problem is solved, if $E f_0(x^*, w)$ is not much longer than $f_0(x^*, Ew)$ and $E f_i(x^*, w)$ is not much longer than $f_i(x^*, Ew)$

then it can be safely assumed that the original problem is solved within some reasonable accuracy

(posterior analysis)

- Variations

Can actually put in between the expectation and the function any convex, non-decreasing function

this allows you to shape all sorts of things you like

in place of $E f_i(x,w) \le 0$ (constraint holds in expectation) can use

$E[f_i(x,w)_+] \le \epsilon$ — the plus function is cvx and non-decreasing; expectation of non-negative part, thus the expected violation

$E[\max_i f_i(x,w)_+] \le \epsilon$ — expectation of worst violation

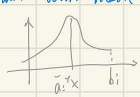
unfortunately, chance constraint $\text{Prob}(f_i(x,w) \le 0) \ge \eta$ (e.g. $\eta = 95\%$) is convex in only a few special cases

e.g. when f_i is affine in x and w (linear constraint with uncertain coefficient)

it's a second-order constraint — EE364A lecture 6

$a_i \sim \mathcal{N}(\bar{a}_i, \Sigma_i)$ — $a_i^T x$ is Gaussian with mean $\bar{a}_i^T x$, variance $x^T \Sigma_i x$

$\text{prob}(a_i^T x \le b_i) = \Phi\left(\frac{b_i - \bar{a}_i^T x}{\|\Sigma^{1/2} x\|_2}\right)$



$\text{prob}(a_i^T x \le b_i) \ge \eta \Rightarrow \Phi\left(\frac{b_i - \bar{a}_i^T x}{\|\Sigma^{1/2} x\|_2}\right) \ge \eta \Rightarrow b_i - \bar{a}_i^T x \ge \Phi^{-1}(\eta) \cdot \|\Sigma^{1/2} x\|_2$

$\bar{a}_i^T x + \Phi^{-1}(\eta) \|\Sigma^{1/2} x\|_2 \le b_i$

socp $\Rightarrow$ change constrained linear programming

Almost-sure constraint

$f_i(x,w) \le 0$ almost sure

$E\, 1\{ f_i(x,w) \le 0\} \le 0$ $\quad 1\{x\} = \begin{cases} +\infty & \text{if } x = \text{false} \\ 0 & \text{if } x = \text{true} \end{cases}$ non-decreasing and convex

$\exp(1/\delta\, f_i(x,w))$ $\quad \exp(1/\delta\, x)$ is cvx and increasing

$\Downarrow$

$E[\exp(1/\delta\, f_i(x,w))]$ $\quad$ expectation preserves convexity

$\Downarrow$

$\log E[\exp(1/\delta\, f_i(x,w))]$ $\quad$ log-sum-exp is convex and non-decreasing

the first term of this the expectation of $f(x,w)$

the second term is the variance, scales with δ

(a general construction for handling risk aversion)

- Expected value of convex function

suppose $F(x,w)$ is convex in x for each w and $G(x,w) \in \partial_x F(x,w)$

$$f(x) = \underset{w}{E}[F(x,w)] = \int F(x,w)\cdot p(w)\, dw \text{ is convex}$$

a subgradient of f at x is $g = \underset{w}{E}[G(x,w)] = \int G(x,w)\cdot p(w)\, dw \in \partial f(x)$

a noisy unbiased subgradient of f at x is $\tilde{g} = 1/M \sum_{i=1}^{M} G(x, w_i)$

w_i are M iid samples

eg. expected value of piecewise linear function

$$\min.\ E[\max_i a_i^T x + b_i]$$

a_i and b_i are random

evaluate noisy subgradient using Monte Carlo method with M samples, and run stochastic gradient method

compare to:

certainty equivalent : $\min.\ \max\{ E[a_i]^T x + E[b_i]\}$ (gives a lower bound)

heuristic : $\min\ \max_i \{ E[a_i]^T x + E[b_i] + \lambda \|x\|_2 \}$

- Online learning and adaptive signal processing

  $(x, y) \in \mathbb{R}^n \times \mathbb{R}$ have some joint distribution
  find vector w s.t. $w^T x$ is a good estimator of y
  choose w to minimize expected value of a convex loss function l

  $$J(w) = \mathbb{E}\, l(w^T x - y)$$

  at each time step, we are given a sample $(x^{(i)}, y^{(i)})$ from the distribution

  noisy unbiased subgradient of J at $w^{(k)}$, based on sample $(x^{(k+1)}, y^{(k+1)})$

  $$g^{(k)} = l'(w^{(k)T} x^{(k+1)} - y^{(k+1)}) \cdot x^{(k+1)}$$

  l' is a subgradient of l
  for $l(u) = u^2$, gives the least-mean-square algorithm
  for $l(u) = |u|$, gives the sign algorithm
  $w^{(k)T} x^{(k+1)} - y^{(k+1)}$ is the prediction error (like neural net training)

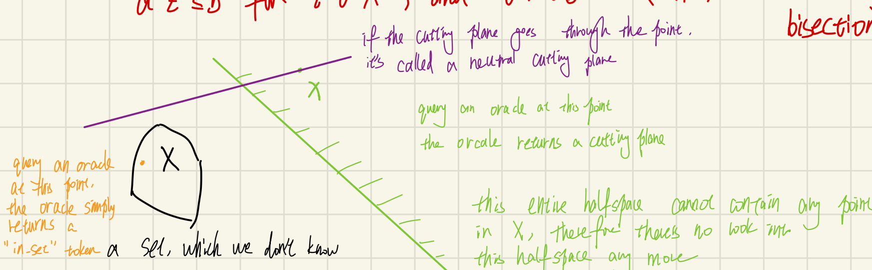
Localization and cutting-plane method are basically bisection in $\mathbb{R}^n$
$\mathbb{R}$, unlike $\mathbb{R}^n$ $(n>1)$, is ordered

- Cutting-plane oracle
  goal: find a point in convex set $X \subseteq \mathbb{R}^n$, or determine that $X = \emptyset$
  our only access to or description of X is through a cutting-plane oracle

  when cutting-plane oracle is queried at $x \in \mathbb{R}^n$, it either
  - assert that $x \in X$
  - returns a seperating hyperplane between x and X

  $$a^T z \leq b \text{ for } z \in X, \text{ and } a^T x \geq b \quad (a \neq 0)$$

  Something like bisection

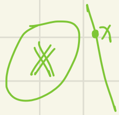


set X can be empty, in which case it can return any plane (conforms the semantic that there's no point in the halfspace that's also in the set)

- Neutral and deep cut

  if $a^T x = b$ (x is on the boundry of halfspace that is cut), cutting-plane is called neutral cut

  if $a^T x > b$ (x is in the interior of halfspace that is cut), cutting-plane is called deep cut



neutral cut

deep cut

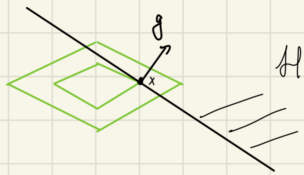
- Unconstrained minimization

  minimize convex function $f: \mathbb{R}^n \to \mathbb{R}$, X is set of optimal minimizers

  given x, find $g \in \partial f(x)$

  from $f(z) \geq f(x) + g^T(z-x)$, we conclude

  $$\underbrace{g^T(z-x) > 0 \Rightarrow f(x) < f(z)}_{\text{get a cutting plane}}$$



$H = \{z \mid g^T(z-x) \geq 0\}$

worse than x, thus not optimal

- Deep cut for unconstrained minimization

  suppose we know a number $\bar{f}$ with $f(x) > \bar{f} \geq f^*$

  (e.g. the smallest value of f found so far in algorithm i.e. f^k_{best}

  $$\bar{f} \geq f^* = f(x^*) \geq f(x) + g^T(x^* - x) \quad g \in \partial f(x)$$

  thus we have a deep cut (x is current point, $\bar{f} = f(x^k_{best})$)

  $$\{z \mid g^T(z-x) + f(x) - \bar{f} \leq 0\}$$

  take $z = x^*$ $\quad g^T(x^*-x) + f(x) \leq f(x^*) - \bar{f} \leq 0$ is in the set

  take $z = x$ $\quad g^T(x-x) + f(x) - \bar{f} = f(x) - \bar{f} > 0$ is not in the set

- Feasibility problem

  find x

  s.t. $f_i(x) \leq 0$ $\quad f_i$: convex

  if x is not feasible, find j with $f_j(x) > 0$, evaluate $g_j \in \partial f_j(x)$

  since $f_j(z) \geq f_j(x) + g_j^T(z-x)$

  $$f_j(x) + g_j^T(z-x) > 0 \Rightarrow f_j(z) > 0 \Rightarrow z \notin X$$

  $\therefore$ any feasible z satisfies the inequality $f_j(x) + g_j^T(z-x) \leq 0$

  take x is not in the half space

  $f_j(x) + g_j^T(z-x) \leq 0$ gives a deep cut

· Inequality constrained problem

$$\min.\ f_0(x)$$
$$\text{s.t.}\ f_i(x) \le 0 \qquad f_0, f_i \text{ convex}, X \text{ is the set of optimal points}$$

if x is not feasible, say $f_j(x) > 0$, we have (deep) feasibility cut

$$f_j(x) + g_j^T(z-x) \le 0 \qquad g_j \in \partial f_j(x)$$

if x is feasible, we have (neutral) objective cut

$$g_0^T(z-x) \le 0 \qquad g_0 \in \partial f_0(x)$$

generate a cutting plane oracle

· localization algorithm

basic (conceptual) localization (or cutting-plane) method

given initial polyhedron $P_0 = \{z \mid Cz \le d\}$ (or any convex set) know to contain X

$k = 0$

repeat

choose a point $x^{(k+1)}$ in P_0

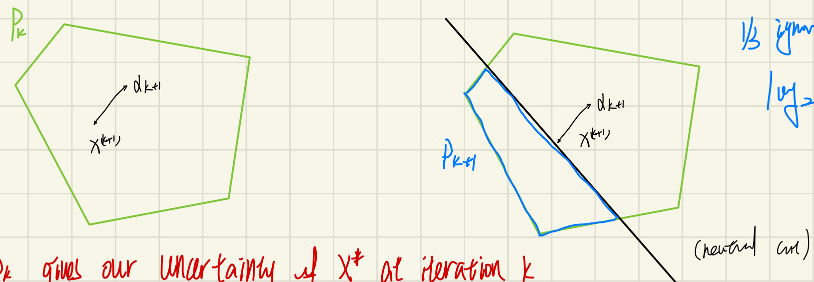
query the cutting plane oracle at $x^{(k+1)}$

if $x^{(k+1)} \in X$, quit (the polyhedron summarizes what you know)

else, add the new cutting plane $a_{k+1}^T z \le b_{k+1}$; $P_{k+1} = P_k \cap \{z \mid a_{k+1}^T z \le b_k\}$

if $P_{k+1} = \phi$ quit, return infeasibility token

$k := k+1$



1/3 ignorance

$\log_2(3) \approx 1.5$

P_k gives our uncertainty of X^* at iteration k

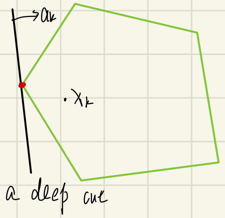
want to pick $x^{(k+1)}$ so that P_{k+1} is as small as possible, no matter what cut is made

want $x^{(k+1)}$ near the center of P_k

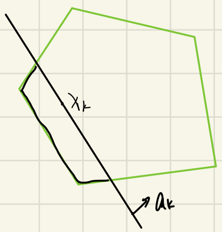
log volume (P_k) is a good measure of ignorance

$\log_2$ volume (P_k) is a good measure of the actual number of bits of information you gain

Boyd intuition time {



query at x_k, best thing can happen is that
it eliminates all of P_k except a vertex
ignorance reduction is too, arbitrarily good thing can happen



ignorance reduced by 1/4
2 full bits gain

ignorance reduction way under
1 bit

}

· Specific cutting-plane methods

center of gravity
maximum volume ellipsoid
chebyshev center
analytic center

· Center of gravity algorithm

take $x^{(k+1)} = \text{Center of Gravity}(P_k)$

$$CG(P_k) = \int_{P_k} x \, dx \Big/ \int_{P_k} dx$$

for any polyhedron, in any dimension,
take the center of gravity of it,
and put a hyperplane going through CG,
the divided volume is never worse than
37 : 63

theorem: if $C \subseteq \mathbb{R}^n$ is convex, $x_{cg} = CG(x)$, $g \neq 0$

$$\text{vol}\left(C \cap \{x \mid g^T(x - x_{cg}) \le 0\}\right) \le (1 - 1/e)\,\text{vol}\,C \approx .63\ \text{vol}(C)$$

can get $\log_2 1/.63$ bit of information

- Convergence of CG cutting plane method

suppose P_0 lies in ball of radius R. X includes ball of radius r
(can take X as set of ϵ suboptimal points)

suppose $x^{(1)}, \dots x^{(k)} \notin X$ so $P_k \supseteq X$

$$\alpha_n \cdot r^n \leq \mathrm{vol}(P_k) \leq (0.63)^k \, \mathrm{vol}(P_0) \leq (0.63)^k \alpha_n R^n$$

α_n is the volume of unitball in R^n

$k \leq 1.51 \cdot n \cdot \log_2 (R/r)$ ($k \leq \log_2 (R/r)$ for bisection)

Advantage of CG-method
- guaranteed convergence
- affine invariance
- number of steps proportional to dimension n, log of uncertainty reduction

Disadvantages
- find $x^{(k+1)} = CG(P_k)$ is much harder than the original problem