Linear feasibility problem

LP primal

min $c^T x$ s.t.

$Ax + b \geq 0$

dual

max $-b^T y$ s.t.

$A^T y = c$, $y \geq 0$

optimal

$\Uparrow$

feasible

find x s.t.

$Ax + b \geq 0$

linear feasibility

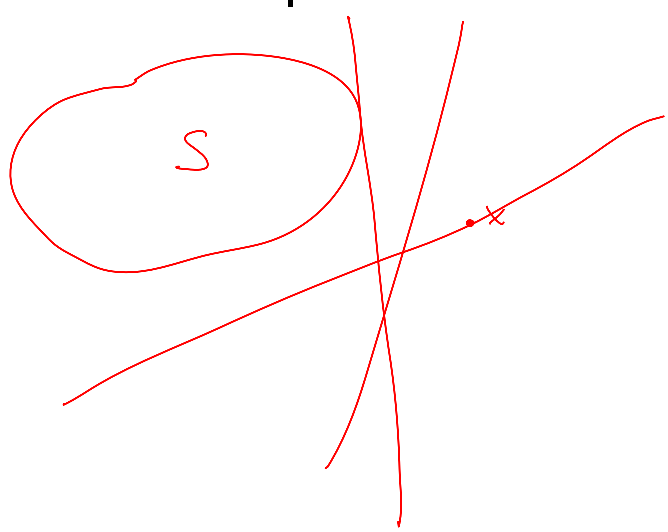
find x, y

primal $\left\{ Ax + b \geq 0 \right.$

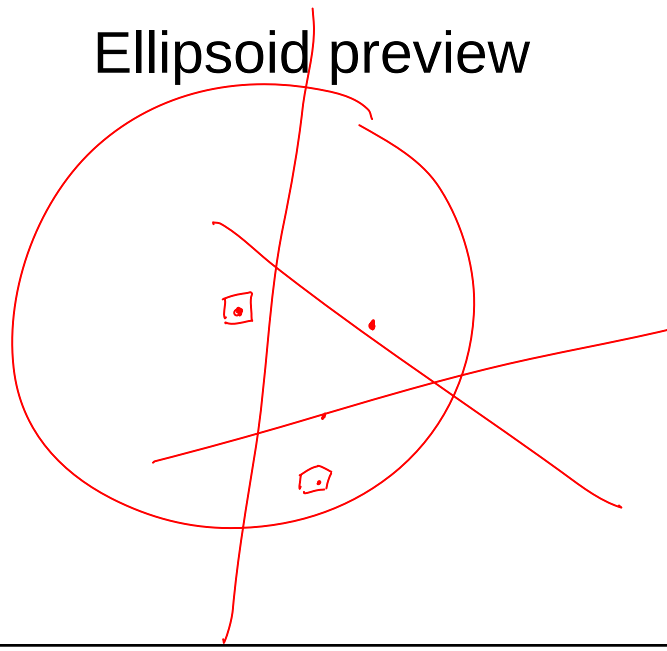
dual $\left\{ \begin{array}{l} A^T y = c \\ y \geq 0 \end{array} \right.$

$c^T x = -b^T y$ ← optimal

Separation oracle



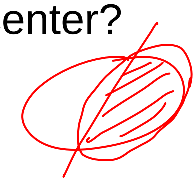
Ellipsoid preview



Difficulties

- How do we get bounding sphere? later
- How do we know when to stop?

  $\forall i \quad a_i^T x + b_i + \eta \geq 0 \qquad \eta > 0$ small const
- Bound region gets complicated–how do we find its center?



Bounding a half-ellipsoid

+ve def

- General ellipsoid w/ center x_C, shape A:

$(x-x_c)^T A (x-x_c) \le 1 \qquad A = U^T U \qquad$ invertible U

- Halfspace: $p^T x \le p^T x_c$ $\longrightarrow p^T(U^{-1}y + \cancel{x_c}) \le \cancel{p^T x_c}$

normal

- Translate to origin, scale to be spherical

$y = U(x-x_c) \qquad x = U^{-1}y + x_c$

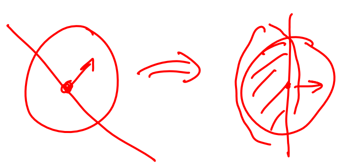
$\underbrace{(x-x_c)^T U^T}_{y^T}\underbrace{U(x-x_c)}_{y} \le 1 \qquad q := U^{-T}p / \|U^{-T}p\|$

$y^T y \le 1 \qquad p^T U^{-1} y \le 0$

$q^T y \le 0$

Bounding a half-sphere

- Rotate so hyperplane is axis-normal



$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

$r = q + e_1 \qquad Rq = e_1$

$R = \frac{2rr^T}{r^T r} - I \qquad R^T R = I$

$z := Ry \Rightarrow z^T z \le 1$

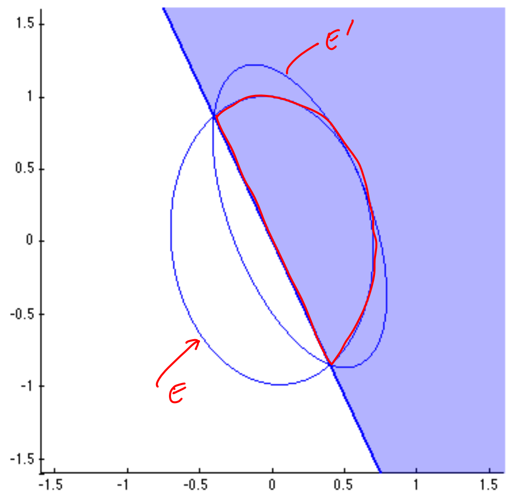
$e_1^T z \le 0$

- New center z_C: $-\frac{1}{n+1} e_1$
- New shape B: $\frac{n^2-1}{n^2}\begin{pmatrix} \frac{n+1}{n-1} & & & 0 \\ & 1 & & \\ & & 1 & \\ 0 & & & \ddots \end{pmatrix}$

$(z-z_c)^T B (z-z_c) \le 1$

$\Leftarrow z^T z \le 1 \wedge e_1^T z \le 0$

For example

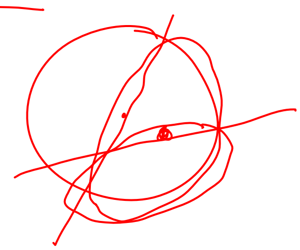


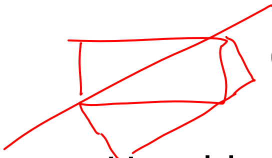
$\text{Vol}(E') \le \text{Vol}(E)\, e^{-\frac{1}{2(n+1)}}$

Ellipsoid algorithm

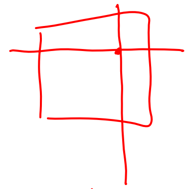
perturbation

- Want to find x s.t. $Ax+b+\eta \geq 0$
- Pick E_0 s.t. $x^* \in E_0$
- for t := 1, 2, …
  - x_t := center of E_t
  - ask whether $Ax_t + b + \eta \geq 0$
    - yes: declare feasible! done!
    - no: get new constraint w/ normal p_t
  - E_{t+1} := bound($E_t \cap \{ x \mid p_t^T x \leq p_t^T x_t \}$)
  - if $\text{vol}(E_{t+1}) \leq \varepsilon \text{vol}(E_0)$: declare infeasible! done!





Getting bounds



- How big does E_0 need to be?
- What should η be?
- How small does ε need to be?

$\log_2 n! 2^{Mn}$
$\approx (n \log n) + Mn$

bit length $\leq$ poly(original bit length)

$B^{-1}b$ B subset of rows of A

Cramer's Rule:
$$(B^{-1})_{ij} = \frac{|B_{\setminus ij}|}{|B|}$$

Laplace's Rule:
$$\forall i \quad |B| = \sum_j (-1)^{i+j} B_{ij} |B_{\setminus ij}|$$

$b \in \mathbb{R}^n$

orig matrix entries $[-2^M, 2^M] \Rightarrow$ new entries $[-n!2^{Mn}, n!2^{Mn}]$

Dotting i's, crossing t's

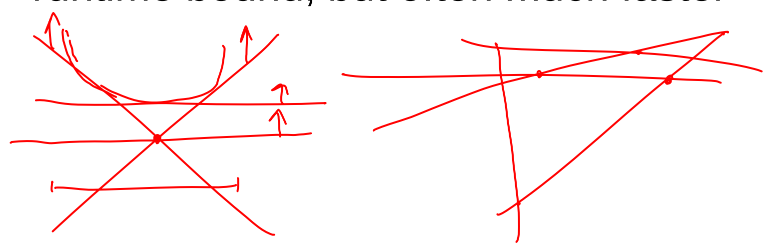
- What if LF was unbounded? So what?
- What about numerical precision?

$A = U^T U \Rightarrow$ square roots $\Rightarrow$ infinite precision !!?

$O(n^3 M)$

Comparison to constraint generation

- Ellipsoid is polynomial, but slow
- Constraint generation has no non-trivial runtime bound, but often much faster



Other algorithms

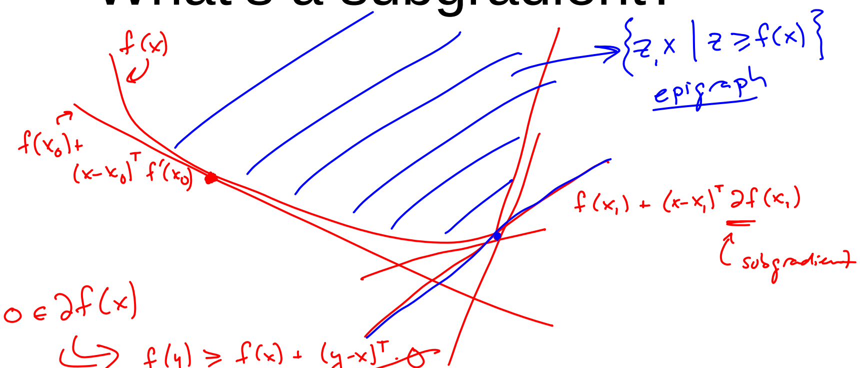
- Interior point: polynomial, can be very fast
- Simplex: exponential in worst case, but often fast in practice

  randomized
- Randomized simplex: polynomial [Kelner & Spielman, 2006]
- Subgradient descent: weakly polynomial, but really simple, and fast for some purposes

  poly$(n, m, M, 1/\epsilon)$

  accuracy

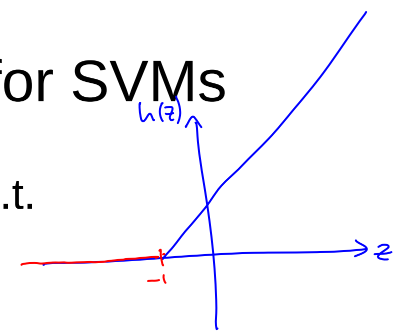
What's a subgradient?



$0 \in \partial f(x)$

$\forall y \hookrightarrow f(y) \geq f(x) + (y-x)^T \cdot 0$

Subgradients for SVMs

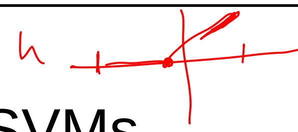


- $\min_{s,w,b}\ \|w\|^2 + C\sum_i s_i$ s.t.

  $\forall i\quad y_i(x_i^T w - b) \geq 1 - s_i$ $\hookrightarrow z_i = y_i(x_i^T w - b)$

  $s_i \geq 0$
- Equivalently, $h(z) = \max\{0, 1+z\}$

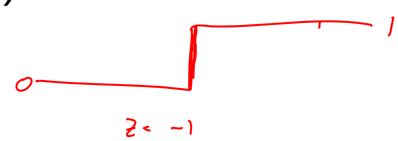
  $\min_{w,b} \|w\|_2^2 + C\sum_i h(-z_i) = \|w\|^2 + C\sum_i h(-y_i(x_i^T w - b))$



Subgradients for SVMs

- $\min_w L(w) = ||w||\text{^}2 + (C/m)\sum_i h(y_i x_i^T w)$
- Subgradient of h(z):

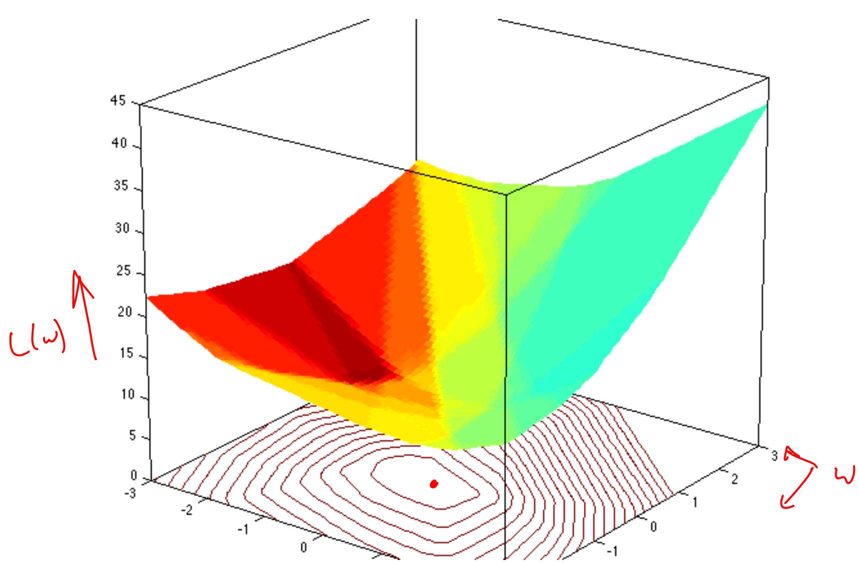
$$\partial(h(z)) = \begin{cases} 0 & z < -1 \\ 1 & z > -1 \\ [0,1] & z = -1 \end{cases}$$



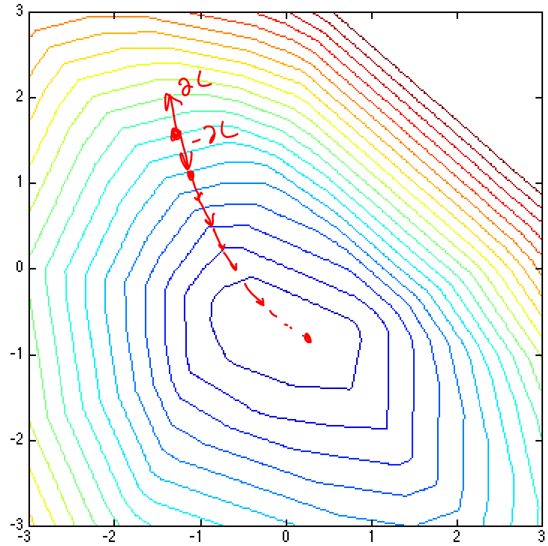
- Subgradient of L(w) wrt w:

$$\partial L(w) = 2w + \frac{C}{m} \sum_i \partial h(-y_i x_i^T w) \cdot (-y_i x_i)$$

SVM loss



Subgradient descent



Subgradient descent

Start w/ x_0

- While not tired:

  g_t = (estimate of) $\partial f(x_t)$

  $x_{t+1} = x_t - \eta_t g_t$

  $x_{t+1} := \Pi_F \, x_{t+1}$

  projection onto feasible region F

η_t = learning rate

Subgradient questions

- How to choose learning rate?
- How to decide when we're tired?
- How to estimate $\partial f(x_t)$?