Linear programming.

- define
- assumptions:
  - nonempty, bounded polyhedron
  - minimizing x_1
  - unique minimum, at a vertex
  - exactly d constraints per vertex
- definitions:
  - hyperplanes H
  - **basis** $B(H)$ of hyperplanes that define optimum
  - optimum value $O(H)$
- Simplex
  - exhaustive polytope search:
  - walks on vertices
  - runs in $O(n^{\lceil d/2 \rceil})$ time in theory
  - often great in practice
- polytime algorithms exist (ellipsoid)
- but bit-dependent (weakly polynomial)!
- OPEN: strongly polynomial LP
- goal today: polynomial algorithms for small d

Random sampling algorithm

- Goal: find $B(H)$
- Plan: random sample
  - solve random subproblem
  - keep only violating constraints V
  - recurse on leftover
- problem: violators may not contain all of $B(H)$
- bf BUT, contain **some** of $B(H)$
  - opt of sample better than opt of whole

- but any point feasible for $B(H)$ no better than $O(H)$
- so current opt not feasible for $B(H)$
- so some $B(H)$ violated

- revised plan:
  - random sample
  - discard useless planes, add violators to "active set"
  - repeat sample on whole problem while keeping active set
  - claim: add one $B(H)$ per iteration

- Algorithm **SampLP**:
  - set S of "active" hyperplanes.
  - if $n < 9d^2$ do simplex ($d^{d/2+O(1)}$)
  - pick $R \subseteq H - S$ of size $d\sqrt{n}$
  - $x \leftarrow$ **SampLP**$(R \cup S)$
  - $V \leftarrow$ hyperplanes of H that violate x
  - if $V \leq 2\sqrt{n}$, add to S

- Runtime analysis:
  - mean size of V at most $\sqrt{n}$
  - each iteration adds to S with prob. 1/2.
  - each successful iteration adds a $B(H)$ to S
  - deduce expect $2d$ iterations.
  - $O(dn)$ per phase needed to check violating constraints: $O(d^2n)$ total
  - recursion size at most $2d\sqrt{n}$

$$T(n) \leq 2dT(2d\sqrt{n}) + O(d^2n) = O(d^2n \log n) + (\log n)^{O(\log d)}$$

(Note valid use of linearity of expectation)

Must prove claim, that mean $V \leq \sqrt{n}$.

- Lemma:
  - suppose $|H - S| = m$.
  - sample R of size r from $H - S$
  - then expected violators $d(m - r - 1)/(r - d)$

- **book broken: only works for empty** S

- Let C_H be set of optima of subsets $T \cup S$, $T \subseteq H$
- Let C_R be set of optima of subsets $T \cup S$, $T \subseteq R$
- note $C_R \subseteq C_H$, and $O(R \cup S)$ is only point violating no constraints of R
- Let v_x be number of constraints in H violated by $x \in C_H$,
- Let i_x indicate $x = OPT(R \cup S)$

$$\begin{aligned} E[|V|] &= E[\sum v_x i_x] \\ &= \sum v_x \Pr[i_x] \end{aligned}$$

- decide $\Pr[v_x]$
  - $\binom{m}{r}$ equally likely subsets.
  - how many have optimum x?
  - let q_x be number of planes defining x **not** already in S
  - must choose q_x planes to define x
  - all others choices must avoid planes violating x. prob.

$$\begin{aligned} \binom{m - v_x - q_x}{r - q_x} / \binom{m}{r} &= \frac{(m - v_x - q_x) - (r - q_x) + 1}{r - q_x} \binom{m - v_x - q_x}{r - q_x - 1} / \binom{m}{r} \\ &\leq \frac{(m - r + 1)}{r - d} \binom{m - v_x - q_x}{r - q_x - 1} / \binom{m}{r} \end{aligned}$$

  - deduce

$$E[V] \leq \frac{m - r + 1}{r - d} \sum v_x \binom{m - v_x - q_x}{r - q_x - 1} / \binom{m}{r}$$

  - summand is prob that x is a point that violates exactly one constraint in r.
    * must pick q_x constraints defining x
    * must pick $r - q_x - 1$ constraints from $m - v_x - q_x$ nonviolators
    * must pick one of v_x violators
  - therefore, sum is expected number of points that violate exactly one constraint in R.
  - but this is only d (one for each constraint in basis of R)

Result:

- saw sampling LP that ran in time $O((\log n)^{O(\log d)} + d^2 n \log n + d^{O(d)}$
- key idea: if pick r random hyperplanes and solve, expect only dm/r violating hyperplanes.

Iterative Reweighting

Get rid of recursion and highest order term.

- idea: be "softer" regarding mistakes
- plane in V gives "evidence" it's in $B(H)$
- Algorithm:
  - give each plane weight one
  - pick $9d^2$ planes with prob. proportional to weights
  - find optimum of R
  - find violators of R
  - if

    $$\sum_{h \in V} w_h \leq (2 \sum_{h \in H} w_h)/(9d-1)$$

    then double violator weights
  - repeat till no violators
- Analysis
  - show weight of basis grows till rest is negligible.
  - claim $O(d \log n)$ iterations suffice.
  - claim iter successful with prob. 1/2
  - deduce runtime $O(d^2 n \log n) + d^{d/2+O(1)} \log n$.
  - proof of claim:
    * after each iter, double weight of some basis element
    * after kd iterations, basis weight at least $d2^k$
    * total weight increase at most $(1 + 2/(9d-1))^{kd} \leq n \exp(2kd/(9d-1))$
  - after $d \log n$ iterations, done.
- so runtime $O(d^2 n \log n) + d^{O(d)} \log n$
- Can improve to linear in n

DNF counting

Rare events

- if p small, huge sample size
- importance sampling biases samples toward event.

Complexity:

- $\sharp\mathcal{P}$-complete.
- PRAS, FPRAS

Coverage algorithm

- given $A_i \subseteq V$, count $\cup A_i$
- problem: random $a \in V$ too rarely satisfies
- Idea: **Bias** sample to create better odds of interesting event
  - work in $\uplus A_i$
  - size known
  - can sample uniformly
  - dense subset of right size
  - "cannonical" assignment is "minimum" copy of assignment for given clause
  - canonical items number same as $\cup A_i$
- Analysis
  - assignment a, satisfies s_a clauses.
  - $\sum_a (s_a/n)(1/s_a)$
  - prob. OK at least $1/m$, so m trials suff.
- unbiased estimator (expectation equals correct value)

Network Reliability