possibly no class monday.

Linear programming.

- definitions:
  - hyperplanes H
  - **basis** $B(H)$ of hyperplanes that define optimum
  - optimum value $O(H)$

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

Key Lemma:

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

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

Define

- m clauses

Complexity:

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

Rare events

- Idea: choose random assignment, count satisfying fraction
- if p small, huge sample size
- importance sampling biases samples toward event.

Coverage algorithm

- given m sets $A_i \subseteq V$, 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 n known
  - can sample uniformly from it
  - dense subset of right size
  - "canonical" assignment is "first" copy of assignment for given clause
  - canonical items number same as $\cup A_i$
- Analysis
  - assignment a, satisfies s_a clauses.
  - $\sum_a (s_a/n)(1/s_a) = m/n$
  - We know n, so can deduce m
  - How many trials needed? Till get $O(\mu_{\epsilon\delta})$ success
  - prob. OK at least $1/m$, so $\tilde{O}(m)$ trials suff.
- unbiased estimator (expectation equals correct value)

Network Reliability