Using LP rounding to design approximation algorithms.

Typically used in context of 0-1 optimization problem. Example: Min Vertex Cover. Given $G = (V, E)$, find the smallest set $S \subseteq V$ such that for every edge $(i, j)$ either $i \in S$ or $j \in S$

Integer programming formulation:

Minimize $\sum_i x_i$

$x_i \in \{0, 1\}$ \forall i

$x_i + x_j \geq 1$ \forall $(i, j) \in E$

LP formulation: relax to $0 \leq x_i \leq 1$ \forall i.

Observe: Optimum value of LP relaxation is a lowerbound on the integer optimum.
Rounding: going from fractional solution to 0-1 soln.

Example: Given fractional solution for Vertex Cover, consider set of all nodes $i$ such that $x_i \geq \frac{1}{2}$.

- Is a vertex cover since $x_i + x_j \geq \frac{1}{2} \Rightarrow$ one of them is $\geq \frac{1}{2}$
- Has size at most $2 \times \sum_i x_i = 2 \times$ fractional opt
- This simple 2-approx is essentially the best we know of for VC !!.

Randomized rounding: make $y_i = 1$ with prob. $x_i$ and 0 with prob. $1-x_i$

Observations (Raghavan-Thompson): 1) For any coefficient vector $a$,

$$E[a \cdot y] = a \cdot x.$$ (Linearity of Expectation)

2) The $y_i$'s are independent random variables, so one can use Chernoff bounds to upperbound the chance that $a \cdot y$ deviates much from the expectation.
3/4 -approx. for MAX-2SAT

Problem: Given 2-CNF formula, find assignment that maximizes number of satisfied clauses.

First we write the LP. Have a variable $x_i$ for each boolean variable $y_i$ and a variable $z_c$ for each clause $c$. Require $0 \leq x_i, z_c \leq 1$

Objective is to maximize $\sum_c z_c$.
If clause $c$ is $y_i \lor y_j$ represent by $x_i + x_j \geq z_c$. (Thus “1” represents “True” and “0” represents “False.”)
If clause is $y_i \lor \neg y_j$ then represent by $x_i + (1-x_j) \geq z_c$, and so on.

Randomized rounding: make $y_i = \text{True}$ with prob. $x_i$

$$\Pr[\text{clause } c \text{ satisfied}] = 1 - (1-x_i)(1-x_j) = x_i + x_j - x_ix_j$$

$$\geq z_c - x_ix_j$$

$$\geq z_c - z_c^2/4 \quad \text{(by AM} \geq \text{GM)}$$

$$\geq \frac{3}{4} z_c$$

So $E[\text{# of clauses satisfied}] \geq \frac{3}{4} \sum_c z_c$
Running time?

- **Method 1**: Repeat poly(n, 1/ε) times; take the best assignment.

  Averaging shows that at each repetition:
  \[
  \Pr[\text{assignment satisfies } > \frac{3}{4}-\varepsilon \text{ fraction of clauses}] \geq 4\varepsilon
  \]

- **Method 2**: Observe that we only use pairwise independence.

  Can do the rounding using pairwise indep. Variables.
  Can exhaustively search through the probability space (recall HW1);
  takes poly(n) time.

  Method 2 gives deterministic algorithm!
Next example: $O(\log n)$-approximation for Set Cover.

(prototype of $O(\log n)$-approx for other problems, eg VLSI wiring)

Problem: Given sets $S_1, S_2, \ldots, S_m$ of $\{1, \ldots, n\}$, find smallest subset $C$ such that $C \cap S_k \neq \emptyset \ \forall \ k$.

LP: $\min \sum_i x_i$

$\sum_{i \in S_k} x_i \geq 1 \ \forall \ k$

$0 \leq x_i \leq 1$

Solve LP. Do randomized rounding.

$\forall k$, $Pr[S_k \text{ gets covered}] = 1 - \prod_{i \in S_k} (1-x_i) \geq 1 - (1-1/S)^S \geq 1 - 1/e$

where $S = |S_k|$.

Now repeat randomized rounding $t$ times and take union of all the sets produced.

$Pr[S_k \text{ still uncovered after } t \text{ reps}] \leq (1/e)^t$.

Making $t = \log_e m + 1$ we see that this prob. is $\leq 1/em$.

$E[\text{size of final set}] = t \times \text{fractional opt.} = O(\log m) \times \text{fractional Opt.}$