Hardness of approximation

Gap problem:

- on input $x$, if $x \notin L$, accept
- if $x$ is "not even close", reject
- does not matter otherwise

Ex:

- Gap-3SAT: On input a 3cnf $\phi$ with $m$ clauses
  - if $\exists a_1, \ldots, a_n$ satisfy all $m$ clauses, accept
  - if $\forall a_1, \ldots, a_n$ clauses satisfied, reject
  $\leq 7/8m$

- $\alpha$-Gap-CLIQUE: On input $<G,k>$
  - if $G$ has clique of size $k$, accept
  - if $G$'s largest clique is of size $\leq \frac{k}{\alpha}$, reject

Optimization:

- on input $x$, output $y$ s.t. $y \geq x$
- $\max: f(x, y) \leq f(x, z)$
- $\min: f(x, y) \geq f(x, z)$

$\alpha$-Approximation:

- on input $x$, output $y$ s.t. $y \geq x$
- $\max: f(x, y) \geq \frac{1}{\alpha} f(x, z)$
- $\min: f(x, y) \leq \alpha f(x, z)$

Thm: If an $\alpha$-approx algorithm (poly-time) exists for $\max$-3SAT, then a poly-time algorithm solving $\alpha$-Gap-3SAT exists as well.

Proof:

- Alg: on input a 3cnf $\phi$ with $n$ variables, $m$ clauses
  - run the $\alpha$-approximation algorithm for $\max$-3SAT
  - get assignment $a_1, \ldots, a_n$ that satisfies
  - $w = \left( \max \text{ possible # of clauses} / \alpha \right)$ clauses
  - If $w \geq \frac{m}{\alpha}$, accept. Else reject
Analysis:
- Runtime is polynomial because \( \alpha \)-alg is polynomial, everything else polynomial.
- Correctness:
  - if \( \phi \) has a satisfying assignment, the \( a_i \), \( i \in \text{returns returned by approx alg satisfied at least} \frac{m}{\alpha} \) clauses, so accept when it should.
  - if \( \phi \) has no assignment satisfying \( \geq \frac{m}{\alpha} \) clauses, then \( a_i \), \( i \in \text{cannot satisfy} \geq \frac{m}{\alpha} \) clauses, so reject when we should.

This is important for showing when \( \alpha \)-approx algorithms are.

Contrapositive of Thm: If \( \alpha \)-Gap-3SAT is \( \text{NP} \)-hard, then no poly-time approx alg exist for MAX-3SAT unless \( P = \text{NP} \).

From PCP thm: can't approximate MAX 3SAT any better than \( \frac{8}{9} \). I.e. no \( \frac{8}{9} - \epsilon \)-approx.

Equivalent thm for clique & for all approx/gap problem:
An \( \alpha \)-approx alg can be turned into an alg solving the corresponding \( \alpha \)-gap problem.

Lab problem 3: gap preserving reduction
from 3SAT to CLIQUE
- On input \( \emptyset \in 3\text{SAT}, \) output \( \leq k \) to CLIQUE
- \( \emptyset \) where can't sat \( \geq \frac{3}{2} + \epsilon \) of clauses
- output \( \leq k \) \( \leq \frac{3}{2} + \epsilon \) of maximum CLIQUE in \( G \)
- \( G \) is of size \( \leq \frac{k}{3} \).

Thm: \( \alpha \)-Gap-CLIQUE is \( \text{NP} \)-hard.
If there is an algorithm solving \( \beta - \text{gap-clique} \), then there is a poly-time algorithm solving \( \sqrt{\beta} - \text{gap-clique} \).

**Corollary:** If for some \( \beta \) there exists a poly-time algorithm for \( \beta - \text{gap-clique} \), then for any \( \varepsilon > 0 \), there is a poly-time alg for \((1 + \varepsilon) - \text{gap-clique}\).

This can be done by taking square roots of \( \beta \) some constant \( \delta \) of times.

**Corollary 2:** There does not exist a poly-time alg for \( \beta - \text{gap-clique} \) for any \( \beta > 1 \) unless \( P = NP \).

**Example graph:** 5 vertices, clique of size 3.

\[
G = (V_G, E_G)
\]

Product graph \( H = G \times G = (V_H, E_H) \)

\[
V_H = V_G \times V_G
\]

\[
E_H = \{(u, v) \mid (u_1, v_1) \in E_G \text{ and } (u_2, v_2) \in E_G \}
\]

A subgraph of \( H \) that makes a clique:

\[
\begin{array}{ccc}
1,1 & 1,2 & 1,3 \\
2,1 & 2,2 & 2,3 \\
3,1 & 3,2 & 3,3
\end{array}
\]

All 9 vertices are connected and form a clique.

\( G \) has maximum clique of size \( k \) iff \( H \) has a clique of size \( k^2 \).
proof cont.

On input $\langle G, k \rangle$,

- construct product graph $H$.
- run algorithm for $\beta$-gap clique on input $\langle H, k^2 \rangle$.
- return whatever it returns.

Analysis: runtime $\sqrt{}$

Correctness: if $\langle G, k \rangle \notin \text{CLIQUE}$, then $\langle H, k^2 \rangle \notin \text{CLIQUE}$, so $\sqrt{}$

If $G$'s largest cliques of size $< \sqrt{\beta} \cdot k $,

- $H$'s largest clique is of size $< (\sqrt{\beta} \cdot k)^2 = \beta \cdot k^2$.

      =>

      so reject $\sqrt{}$, so reject.

Therefore with a $\sqrt{\beta}$-gap clique algorithm, it can solve a $\sqrt{\beta}$-gap clique

in poly time.
**Probabilistic algs**

BPP, RP, coRP, ZPP

![Diagram showing the relationships between BPP, RP, NP, PSPACE, coRP, ZPP, and P.]

Possible: $P \neq BPP$ (weaker)

$ZPP \neq BPP$

$L \leq BPP$ if for some ppt TM $M$

$\forall x \in L \Rightarrow M \text{ accepts } x \text{ w. prob. } \geq \frac{2}{3}$

$\forall x \not\in L \Rightarrow M \text{ accepts } x \text{ w. prob. } \leq \frac{1}{3}$

**Thm (Amplification of BPP):**

If $L \leq BPP$, can construct a ppt TM $M'$ so

$\forall x \in L, M' \text{ accepts } x \text{ w. prob. } \geq 1 - 2^{-|x|}$

$\forall x \not\in L, M' \text{ accepts } x \text{ w. prob. } \leq 2^{-|x|}$

Do $T$ times and take majority answer, reduce possibility of error.