15854(S): (Advanced) Approx Algs

- HW5 (10 weeks)
- Pizza / Geocope / etc.

- Optimization Problems vs. Decision Problems
  - 3SAT vs. Max 3SAT, min 3-unSAT etc.
  - 2SAT (in P) vs. Max 2SAT, min 2-unSAT
  - 3color (NP-complete) vs. Min-Coblog, Max 3 cut etc.
    2color (in P) vs. Max 2cut, Min 2 uncut.

- Landscape is Approx Problem

  - Input: instance \( I \).
  - Valid solution \( S \in \text{sols}(I) \).
    Assume: checking \( I \in \text{NP} \).
    Also, that there exists at least
    non-trivial value for each \( S \in \text{sols}(I) \).
  - min or max.

  Want: algo: starts searches to solution st. \( \text{val}(S(I)) \) is as high as possible.

Example: Max 3SAT. Instance one 3SAT formula (\( m \) clauses, \( n \) vars)

- Solutions: any truth assignment
- Assignment of T/F to each var.
- Value: \( \frac{\text{# satisfied clauses}}{\text{total # clauses}} \) \( \in [0,1] \).

Also: assign random T/F assignment.

Produces solution that satisfies \( \frac{2}{3} \) clauses in expectation.

Approx ratio: worst case value \( \frac{\text{Alg}(I)}{\text{OPT}(I)} \). Can we do better?
Finer grained Landscape than just NP-hardness

1. \text{Min-TSP (R^2)} is NP hard. but has PTAS. — later
2. \text{Min-Set cover} is NP-hard but has \(O(\log n)\) apx. — soon
3. \text{Max 2SAT} — but \(\frac{7}{8}\) apx is best possible
4. \text{Max Clique} — \(n^{\frac{1}{2}}\) apx \(\Rightarrow P=NP\).

So goal of course: apx algorithms (show positive results)

hardness of apx (negative results)
or integrality gaps / algorithmic gaps (limitations of our algorithms).

Sometimes talk about c-vs-s search problems. \(s \leq c\)

- if \(OPT(I) \geq c\) produce soln w/ value \(\geq s\)

- or c-vs-s decision problems.

- if \(OPT(I) \geq c\), answer \text{YES}, else \text{if} \(OPT(I) < c\), answer \text{NO}

(anything in the middle is OK)

Fact: \text{Algorithm for c-vs-s SEARCH } \Rightarrow \text{c-vs-s DECISION}

\text{If: run alg. if } OPT(I) \geq c \text{ then } \text{ALG}(I) \Rightarrow \text{YES} \text{ (so output YES if } \text{ALG}(I) > c)\text{)

if } \text{ALG}(I) \Rightarrow \text{ OPT(I) } \leq \text{ so answer NO is OK.}

\text{Run alg. if } \text{ALG}(I) \geq c \text{ say YES, else say NO.}

\text{OPT}(I) \geq c \Rightarrow \text{YES is fine.}

\text{OPT}(I) < c \Rightarrow \text{NO is fine.}
\[ \min \text{Set cover: General Problem, captures many other. (HW?)} \]

**Input.** Set system \( F = \{ U, S_1, S_2, \ldots, S_m \} \) \( S_i \subseteq U \).

Assume \( \bigcup_{i=1}^{m} S_i = U \).

Sets have cost \( c_1, \ldots, c_m \geq 0 \) \( \text{(sometimes unit cost in interest as well)} \).

**Solution.** Sub collection \( I \subseteq \{ 1, \ldots, m \} \)

\[ \text{st } \bigcup_{i \in I} S_i = U. \]

**Value.** \( \text{cost } / \text{cardinality} \).

\[ \text{with S.C. } \quad \text{unweighted S.C.} \]

\[ \text{cost}(I) = \sum_{i \in I} c_i \]

\[ |I| \]

---

**Algorithm Greedy:** (unweighted)

\[ \text{while not all elements covered} \]

\[ \quad \text{pick set } S_i \text{ that covers most uncovered elements.} \]

**Thm.** Greedy is a \((\ln n)\)-approximation for unweighted set cover.

**Pf.** \( \text{Spr OPT}(I) = K \). \( \text{then Spr } n_t = \# \text{uncovered elt. after } k \text{ rounds.} \)

\[ n_0 = n. \]

**Fact.** If set has covers \( \geq \frac{1}{K} \text{ frac of current uncovered elt.} \)

**Pf.** \( \text{OPT covers the } n_t \text{ elt. so I set in OPT that covers } \geq \frac{n_t}{K} \)

\[ n_{new} \leq n_t \left( 1 - \frac{1}{K} \right) \Rightarrow n_t \leq n_0 \left( 1 - \frac{1}{K} \right)^T < n_0 e^{-T/K} = 1 \text{ elt.} \]

\[ \text{if } T = k \ln n. \]
Fact: Greedy solves the 1-vs- (1/2) for Max-k-coverage.

\[ \text{Oh: Max-k-coverage. Solutions } = K \text{ sets in set system} \]
\[ \text{Val } = \# \text{ elements covered by } K \text{ sets.} \]

Pf: After K picked, \( n_k \leq n_0 (1 - \frac{1}{k})^k < n_0(1/e) \cdot n^{-1} \)
\[ \Rightarrow \text{coverage } > \frac{n(1-1/e)}{n} = 1 - \frac{1}{e}. \]  

---

Fact: if \( \text{Gap also for Max-k-coverage } \text{ and in 1-vs-(1/e) search also.} \)
\[ \Rightarrow \exists \left( \log (1/e) n^{+1} \right) \text{ gap also for min set cover.} \]

Pf: "Guess" \( \text{OPT}=K. \)

Use also \( A \). Covers 1-\( g \). Repeat on remainder.
\[ \min K \]
\[ \Rightarrow \# \text{ uncovered after } T \text{ rounds } = \sum_{i=T}^{\infty} n < 1 \]
if \( T = \log (1/e) n^{+1} \).

---

Corollary: 1-vs-(1-1/e) for Max-k-Cov \( \Rightarrow \ln n \) for Min Set Cover

**Remark:** if \( \ln n \) (1-e) for SC hard \( \Rightarrow (1-1/e+8) \) for Max-k-Cov is hard.

a little more complicated, maybe see.  
see this in a future lecture.
Algorithmic Gap: Does greedy do better than \( \ln n \)?

Fact: Greedy no better than \( \ln n (1-\varepsilon) \) (even for unweighted)

PF:

\[
\begin{align*}
\text{OPT} &= 2. \\
\text{OPT} &= \log n (1/2) \\
\Rightarrow \text{Gap} &= \log \left( \frac{n}{2} \right) \\
&= \frac{\ln n - 1}{2} \\
&\leq \frac{\ln n}{2 \ln 2}.
\end{align*}
\]

To set \( \ln n \), use set coverage \( \text{OPT} = K \) vertically.

But each other set covers \( \frac{1}{K} \) of remainder.

\[
\Rightarrow \text{#sets} = \log \left( \frac{1}{1-K} \right) n. \\
\Rightarrow \text{Gap} = \log \left( \frac{1}{1-K} \right) n \\
= \frac{\ln n}{K \ln \left( \frac{1}{1-K} \right)}
\]

but \( \ln(1+\varepsilon) = \varepsilon + \Theta(\varepsilon^2) \)

for \( \varepsilon \) small.

So another algorithm?

Before that: am greedy algo for weighted case.

at each step, pick set that \( \max \left( \frac{\text{coverage}}{\text{cost}} \right) \)

Thus: Greedy is \( \Theta \ln n \) - apx for weighted set cover.

PF: (sketch) Same ideas as before. Show that if cost \( c_1, c_2, \ldots, c_k \)

\[
\begin{align*}
\text{cost}_t &\leq n \left( 1 - \frac{c_1}{\text{OPT}} \right) \left( 1 - \frac{c_2}{\text{OPT}} \right) \cdots \left( 1 - \frac{c_k}{\text{OPT}} \right) \\
&\leq n \exp \left( \frac{\sum c_i}{\text{OPT}} \right) \\
&\leq \text{etc}
\end{align*}
\]
Linear Program - based Algos:

**Idea:** Relax-and-Round

1. Write an IP for Set Cover. (IP = Integer (Linear) Program).
2. "Relax" it to an LP (LP = Linear Program).
3. Solve this LP. 
4. "Round" the fractional solution to integers.

Usually:
1. $IP(I) = Opt(I)$. 
2. $LP(I) \leq IP(I)$. 
3. $Alg(I) \leq \alpha \cdot LP(I)$ \quad $\Rightarrow$ \quad $Alg(I) \leq \alpha \cdot Opt(I)$.

**Set Cover:** variable $x_s \in \{0, 1\}$ for each set $S \in \{S_1, S_2, \ldots, S_m\}$.

**IP:**

$$\min \sum x_s$$

$$st \sum_{S \in U} x_s \geq 1 \quad \forall \epsilon U.$$  

$x_s \in \{0, 1\}$.  

$x_s \geq 0$

**LP:**

Round: Imagine each $x_s$ as a prob. value. (Fact: $x_s \in \{0, 1\}$, no reason for $x_s$ to be larger).

**Alg:**

For $T$ times

$$\forall S \in F$$

select $S$ independently w.p. $x_s$. T rounds sampling
What if this is not a feasible solution?

Clean-up: Eliminate $e$, pick cheapest set covering $e$ if $e$ not covered by sample.

**Lemma:** $E[\text{cost of solution}] \leq T \cdot LP(D) + \left[ \sum_{\text{cell}} (\text{cheapest set covering}) \right] e^{-T}$.

If $E[\text{cost of each round}] = \sum_{s} C_{s}$. $Pr[\text{picked } s] = \sum_{s} C_{s} x_{s} = LP(D)$.

Now, $\Pr(e \text{ not covered}) = \prod_{s} (1 - x_{s}) \leq e^{-\sum_{s} x_{s}} \leq e^{-1}$

in one round $s$: ees

$\Rightarrow \Pr(e \text{ not covered in } T \text{ rounds}) \leq e^{-T}$.

Now use linearity of expectation again.

Hence set $T = \ln n$.

$E[\text{cost}] \leq (\ln n) \cdot LP + \frac{1}{e} \cdot x_{e} \cdot LP$

$= (\ln n + 1) LP.$

$b/c \text{ LP value } \geq \text{ cheapest set covering for any } e$

HW: Show that if sets are $\Omega(B)$, then LP roughly gives $O(ln B)$ apx.

Greedy too (but see more later).

**Picture**

- $OPT(E)$
- $LP(D)$
- $IP(D)$
- $A(S(D))$

$\leq \ln n \Rightarrow \text{ increase}$
Ask 2 questions:

1. **Algorithmic gap**: does I instance where 
\[
\frac{As(I)}{OPT(I)} = \Omega(\log n).
\]

2. **Inequality gap**: does I instance s.t. 
\[
\frac{OPT(I)}{LP(I)} = \Omega(\log n).
\]

shows that using this approach cannot beat the log-approximation, no matter what randomization we do.

[as long as we relate ourselves to the LP value, of course!]

**Also gap**: see in HW.

**Inequality gap**:

- Take \( U = \frac{d}{2} \cdot U \leq \frac{d}{2} \cdot |U| \cdot d^{1/3} \) \( d \leq \log_2 n. \)

- \( n = |U| = \left( \frac{d}{2} \right)^{1/3} \Omega \left( \frac{d^2}{\sqrt{d}} \right). \)

- \( \text{Sets: all "dictator" sets } S_k = \frac{d}{2} \cdot U / \forall \in S_k \; \text{cost}=1. \)

- \( \text{OPT} \geq \frac{d}{2}+1 \) else \( f \) element not covered.

- \( \text{LP value: set } \frac{d}{2} \text{ on each set } S_k. \) \( (i.e. \ x_k \leq 1 \ + S). \)

\[ \Rightarrow \text{total LP value} = d \cdot \frac{d}{2} = 2. \]

\[ \Rightarrow \text{Inequality gap} = \frac{d}{2}+1 = \Omega \left( \frac{d^2+1}{2} \right) = \Omega(d) = \Omega(\log n). \]

**Fact**: Can do better, get \( \log n \) for inequality gap as well.