Maximum Coverage

$\mathcal{U}$= set of n elements

$S_1, S_2, \ldots, S_m$ subsets of $\mathcal{U}$

integer $k \leq m$

Goal: choose k sets to maximize number of elements covered

Greedy Algorithm

Mark all elements in $\mathcal{U}$ as uncovered

repeat

  pick set that covers max # of *uncovered* elmts

  mark elements in chosen set as covered

until *done*

Set Cover: *done* – all elements covered

Max Coverage: *done* – k sets picked

Analysis for Max Coverage

Theorem: Greedy is a $(1-1/e) \simeq 0.632$ approximation for max coverage

Proof:

x_i : number of new elements covered in iter. i

y_i : total number of elements covered in iters 1 to i

$y_0 = 0$, y_k - number of elements covered by Greedy

z_i : $OPT - y_{i-1}$

$z_0 = OPT$

Lemma: $z_i \leq (1-1/k)^i \, OPT$

Analysis for Max Coverage

Lemma: $z_i \leq (1-1/k)^i$ OPT

Proof by induction on i

Claim: $x_i \geq z_{i-1}/k$ (Why?)

Therefore

$z_i = z_{i-1} - x_i \leq z_{i-1} (1 - 1/k) \leq (1-1/k)^i$

Analysis for Max Coverage

Lemma: $z_i \leq (1-1/k)^i$ OPT

Therefore $z_k \leq (1-1/k)^k$ OPT $\leq 1/e$ OPT

Greedy covers y_k = OPT - $z_k \geq (1-1/e)$ OPT

Remark: same analysis works if each element $e \in \mathcal{U}$ has a weight $w(e)$ and the goal is to pick k sets that maximize the weight of the elements covered