Set Cover

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

$S_1, S_2, \ldots, S_m$ subsets of $\mathcal{U}$ s.t $\cup_i S_i = \mathcal{U}$

$S_1, S_2, \ldots, S_m$ together *cover* $\mathcal{U}$

Goal: choose as *few* sets as possible to cover $\mathcal{U}$

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

Analysis for Set Cover

Theorem: Greedy is an $(\ln (n/OPT) + 1)$ approximation algorithm for Set Cover

Corollary: Suppose $|S_i| \leq d$ for $1 \leq i \leq m$

Greedy gives a $(\ln d + 1)$ approximation
Since $OPT \geq n/d$ (why?)

Note that OPT here refers to the number of sets (unlike number of elements in Max Coverage)

Analysis for Set Cover

Theorem: Greedy is an $(\ln (n/OPT) + 1)$ approximation algorithm for Set Cover

let $k^* = OPT$

z_i : number of elements uncovered after i sets

t : number of sets picked by Greedy

From previous max coverage analysis after i steps
$z_i \leq (1-1/k^*)^i$

After $j = \ln (n/k^*)\ \ k^*$ iterations
$z_j \leq k^*$

Analysis for Set Cover

Theorem: Greedy is an $(\ln (n/OPT) + 1)$ approximation algorithm for Set Cover

t : number of sets picked by Greedy

From previous max coverage analysis after i steps
$z_i \leq (1-1/k^*)^i$

After $j = \ln (n/k^*)\ k^*$ iterations
$z_j \leq k^*$
$t \leq j + k^*$ (why?) $\leq \ln (n/k^*)\ k^* + k^*$
$\leq (\ln (n/k^*) + 1)\ k^*$

(Near) Tight Example

Consider disjoint sets

$U_1, \ldots, U_k$ with $|U_i| = 2^{i-1}$

$U'_1, U'_1, \ldots, U'_k$ with $|U'_i| = 2^{i-1}$

$\mathcal{U} = \uplus_i U_\rangle \cup \uplus_i U'_i$, $|\mathcal{U}| = 2^{k+1}$

$R_1 = \uplus_i U_i$

$R_2 = \uplus_i U'_i$

for $0 \leq i \leq k$, $C_i = U_i \cup U'_i$

Set cover instance: $\mathcal{U}, R_1, R_2, C_1, \ldots, C_k$

(Near) Tight Example

$\mathcal{U} = \uplus_i U_i \cup \uplus_i U'_i$

$R_1 = \uplus_i U_i$

$R_2 = \uplus_i U'_i$

for $0 \leq i \leq k$, $C_i = U_i \cup U'_i$

Set cover instance: $\mathcal{U}$, R_1, R_2, C_1, …, C_k

OPT = 2 (R_1 and R_2)

Greedy picks C_k, C_{k-1}, …, C_1

hence ratio = $k/2 = \Omega(\log n)$

Set Cover: weighted version

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

$S_1, S_2, \ldots, S_m$ subsets of $\mathcal{U}$ s.t $\cup_i S_i = \mathcal{U}$

$c_1, c_2, \ldots, c_m$ non-negative weights/costs of sets

Goal: choose as *minimum cost* collection of sets to cover $\mathcal{U}$

Greedy Algorithm

density of set $S_i = c_i/|S_i|$

$C = \emptyset$ // covered elements

repeat

  $j = \min_k c_k/|S_k - C|$

  add S_j to solution

  $C = C \cup S_j$

until $C = \mathcal{U}$

Greedy Algorithm

Theorem: Greedy is a H_n approximation algorithm

Proof:

$\mathcal{U} = \{e_1, e_2, \ldots, e_n\}$

Wlog assume elements are covered in order

$e_1, e_2, \ldots, e_n$ (ties resolved arbitrarily)

Assign values to elements: $p : \mathcal{U} \rightarrow \mathcal{R}^+$

Analysis

Wlog assume elements are covered in order

$e_1, e_2, \ldots, e_n$ (ties resolved arbitrarily)

Assign values to elements: $p : \mathcal{U} \rightarrow \mathcal{R}^+$

s.t

$\sum_i p(e_i)$ = cost of all sets picked

and

$p(e_i) \leq OPT/(n-i+1)$ for $1 \leq i \leq n$

Analysis

$\sum_i p(e_i)$ = cost of all sets picked

and

$p(e_i) \leq OPT/(n-i+1)$ for $1 \leq i \leq n$

total cost of sets picked = $\sum_i p(e_i)$

$\leq OPT\ (1/n + 1/(n-1) + \ldots + 1)$

$\leq OPT\ H_n$

Analysis

$\sum_i p(e_i)$ = cost of all sets picked

and

$p(e_i) \leq OPT/(n-i+1)$ for $1 \leq i \leq n$

h_j : index of set picked in iteration j

C_j: elements covered in iterations 1 to $j-1$

$S'_{h_j} = S_{h_j} - C_{j-1}$: new elements covered by S_{h_j}

Analysis

h_j : index of set picked in iteration j

C_j: elements covered in iterations 1 to j-1

$S'_{h_j} = S_{h_j} - C_{j-1}$: elements covered by S_{h_j}

For each e in S'_{h_j} set $p(e) = c_{h_j} / |S'_{h_j}|$

So $\sum_{e \in \mathcal{U}} p(e) =$ total cost of sets picked

Analysis

$p(e_i) \leq OPT/(n-i+1)$ for $1 \leq i \leq n$

Let e_i be covered in iteration j

implies $e_i, e_{i+1}, \ldots, e_n$ not in C_{j-1}

There is a solution of cost OPT that covers these elements

$\Rightarrow$ there is a set of density $OPT/(n-i+1)$

hence $c_{h_j}/|S'_{h_j}| \leq OPT/(n-i+1)$

therefore $p(e_i) \leq OPT/(n-i+1)$

Tighter analysis

$d = \max_j |S_j|$

Greedy's approximation ratio is H_d even for weighted case

Proof: Homework

Oracle access to Sets

Often in applications m, the # of sets is exponential in n or even infinite and is given only implicitly

Greedy algorithm can still be applied provided there is an *oracle* that gives the best density set in each iteration

Approximate oracle

Suppose the oracle is *approximately* good set in each iteration?

That is, say d^* is the density of the smallest density set available to cover remaining elements

Oracle *guarantees* to return a set S such that $d(S) \leq \alpha\, d^*$ for some $\alpha > 1$

α: approximation guarantee of oracle

What can we say about resulting algorithm?

Approximate oracle

Oracle *guarantees* to return a set S such that $d(S) \leq \alpha\, d^*$ for some $\alpha > 1$

α: approximation guarantee of oracle

What can we say about the quality of the resulting solution?

Prove:

Set cover: $\alpha\, H_n$

Max coverage: $1 - e^{-1/\alpha}$

Applications

Vertex Cover

Dominating Set and bounded radius facility location

Multiple Knapsack with identical bin capacities

Vertex Cover

Given graph $G = (V, E)$

$S \subseteq V$ is a *vertex cover* for G iff for each edge $e \in E$, at least one end point of e is in S

Goal: find vertex cover of smallest size

Weighted version: weights on vertices $w: V \rightarrow \mathcal{R}^+$

Goal: find minimum weight vertex cover

Vertex Cover

VC is a special case of Set Cover (why)

Hence $O(\log n)$ approximation for both weighted and unweighted (cardinality) versions

Exercise: construct a graph where Greedy algorithm gives a solution that is $\Omega(\log n)$ OPT

Can do better for VC: 2-approx known

Vertex Cover

2-approx for cardinality VC:

Let M be any *maximal matching* in G (can find a maximal matching in polynomial time)

Claim: OPT $\geq$ |M|

Let S(M) = {u, v | uv \in M }

Claim: S(M) is a VC

Therefore |S(M)| = 2|M| $\leq$ 2OPT

Vertex Cover: weighted version

2-approx via

1. LP rounding
2. Primal-dual

Later

Set Cover with small frequencies

VC is an instance of SC where each element is in (exactly) at most two sets

Given an instance of SC
for $e \in \mathcal{U}$, $f(e)$ = number of sets containing e

$f = \max_e f(e)$ the maximum frequency

For SC, can obtain a f-approximation using LP rounding/primal dual (later classes)

Dominating Set

Given $G=(V,E)$, find a smallest dominating set in G where S is a dominating set if for every $v \in V$,
either $v \in S$ or some neighbour u of v is in S

Exercise: show that DS is a special case of Set Cover

Exercise: show that Set Cover can be essentially reduced to DS

Multiple Knapsack Problem (MKP)

Knapsack problem:

given n items and a knapsack of capacity B

item i has size s_i and profit p_i

Goal: find a maximum profit subset of items that can fit in the knapsack capacity

Multiple Knapsack: given items and m knapsacks each of capacity B

Goal: pack items into knapsacks to maximize profit

Greedy algorithm for MKP

Suppose we have an \alpha approximation algorithm for the single knapsack problem

Greedy for MKP:

for i = 1 to m do

pack the i'th bin using the single knapsack approx algorithm using remaining items

remove items packed in the knapsack

endfor

Greedy algorithm for MKP

Exercise: show that Greedy for MKP is a $1-e^{-1/\alpha}$ approximation by the following

1. show that MKP can be cast as a maximum coverage problem with an exponential sized set system
2. show that the greedy algorithm for mkp is essentially the greedy algorithm for max coverage with the single knapsack algorithm as the desired approximate oracle

Submodular set functions and Greedy

Greedy algorithms work more generally for *submodular set* functions

E a finite set

f: $2^E \rightarrow \mathcal{R}^+$ is submodular iff

$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq E$

Submodular set functions and Greedy

f: $2^E \rightarrow \mathcal{R}^+$ is submodular iff

$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq E$

Another characterization of submodularity (easier to check)

$f(A+i) - f(A) \geq f(B+i) - f(B)$ for all i and $A \subset B$

(decreasing utility)

Submodular set functions and Greedy

f: $2^E \rightarrow \mathcal{R}^+$ is submodular iff

$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq E$

or

$f(A+i) - f(A) \geq f(B+i) - f(B)$ for all i and $A \subset B$

f *monotone* if $f(A+i) \geq f(A)$ for all i, A

(can assume $f(\emptyset) = 0$)

Examples of submodular functions

- set systems
- from matroids (rank functions etc)
- cut functions in graphs (not monotone)
- entropy of random variables
- ...

Example: coverage of set systems

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

$E = \{1, 2, \ldots, m\}$

$f: 2^{\mathcal{E}} \rightarrow \mathcal{R}^+$

defined by
$f(A) = | \cup_{i \in A} S_i |$ for $A \subseteq E$

f is submodular

Example: coverage of set systems

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

each $x \in \mathcal{U}$ has a non-negative weight $w(x)$

$E = \{1, 2, \ldots, m\}$

$f: 2^{\mathcal{E}} \rightarrow \mathcal{R}^+$

defined by
$f(A) = w(\cup_{i \in A} S_i)$ for $A \subseteq E$

f is submodular

Two problems

For *monotone* submodular functions

- Submodular Maximization
- Submodular Set Cover

Generalize Maximum Coverage and Set Cover

Maximizing a Submodular set func

Given E, $f:2^E \rightarrow \mathcal{R}^+$ and integer $k \leq n$

$\max_{S \subset E} f(S)$

s.t

$|S| \leq k$

f is specified via an *oracle*: given a set $A \subseteq E$ the oracle will return f(A)

Submodular cover problem

Given E, $f:2^E \rightarrow \mathcal{R}^+$ and $c: E \rightarrow \mathcal{R}^+$

$\min_{S \subseteq E} c(S)$

s.t

$f(S) = f(E)$

Greedy algorithm generalizes

for submodular coverage

$S = \emptyset$

for $i = 1$ to k do
  pick $e \in E - S$ s.t $f(S+e) - f(S)$ is maximized

  $S = S + e$

endfor

Output S

Greedy algorithm generalizes

for submodular set cover

$S = \emptyset$

while $f(S) \neq f(E)$ do
  pick $e \in E - S$ s.t $f(S+e) - f(S))/c(e)$ is maximized

  $S = S + e$

endwhile

Output S

Performance of Greedy

$1 - 1/e$ for submodular coverage

H_d for submodular set cover where
$d = \max_{i \in E} f(i)$

Note: we are assuming that f is *monotone*

Application

Homework