Optimization with uniform size queries

Uriel Feige

Weizmann Institute

Joint work with Moshe Tenneholtz

Work done at Microsoft Research, Herzeliya

Submodular functions and conventions

U a universe of n items.

f a set function.

Submodularity (decreasing marginals):

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$$

$f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ whenever $S \subset \mathrm{T}$.

Conventions:

Non-negativity $f(S) \geq 0$.

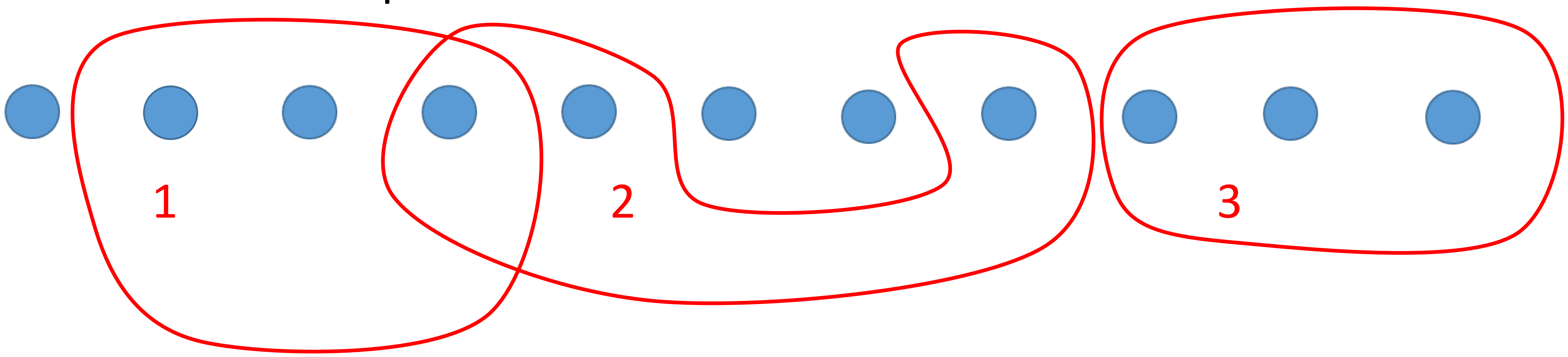
Monotonicity $f(T) \geq f(S)$ whenever $S \subset \mathrm{T}$.

Coverage functions

A special case of non-negative monotone submodular functions.

Each item is a set of elements.

The value of a set of items is the number of elements covered by the union of their respective sets.



Max k-submodular

Given universe U, submodular function f and a parameter k, output a set S of cardinality k that maximizes f(S).

NP-hard.

Approximable within a ratio of $1 - \frac{1}{e}$ [Nemhauser, Wolsey, Fisher 1978], and value queries suffice for this.

NP-hard to approximate within ratios better than $1 - \frac{1}{e}$ [Feige 1998], even in the special case of max k-coverage.

Access to submodular functions

Explicit representation (truth table) – 2^n entries. Exponential.

Implicit representation. For example, coverage functions in the special case in which the number of elements is polynomial in n.

Value query model: an oracle that given S returns f(S).

If the oracle is a polynomial time algorithm, then the oracle algorithm can serve as an implicit representation.

However, value query models capture situations that are more general than implicit representations.

Value queries as physical queries

Universe U – pool of workers for some repetitive task (e.g., picking fruit).

Function f – throughput (e.g., amount of work completed in one day).

(Submodularity – workers distract each other in various ways.)

What is the best set of k workers to hire?

A value query – try a group of workers and observe their throughput.

No implicit representation for f is available.

Value queries as physical queries

Universe U – products that can be displayed in a shop window.

Function f – fraction of those people who pass by the shop who choose to enter the shop (e.g., within one day).

(Submodularity – a passer by would enter the shop even if he/she likes just one of the products displayed in the shop window.)

What is the best set of k products to display?

A value query – place a set of products and observe what happens.

No implicit representation for f is available.

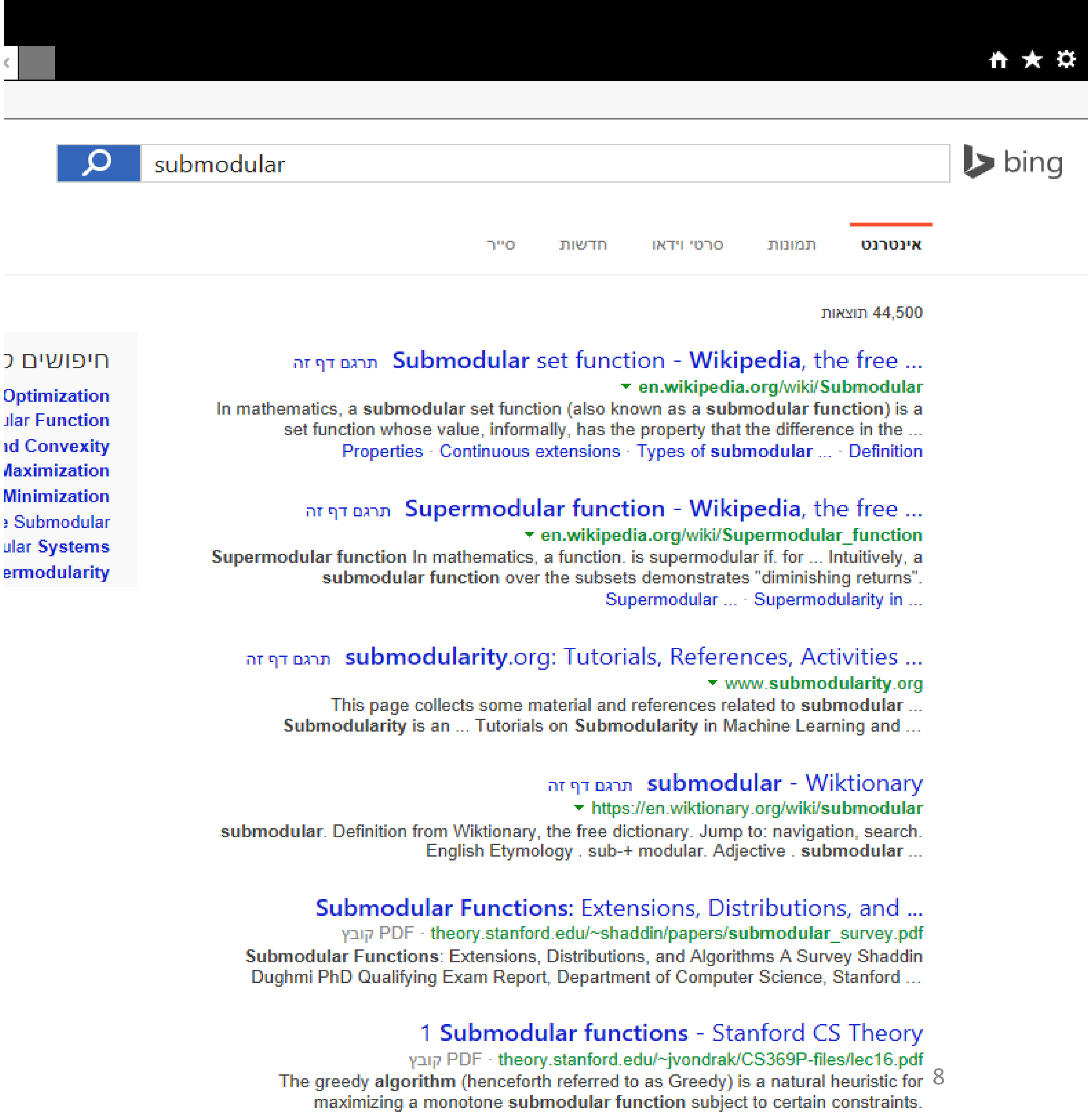
Universe U – Potential replies to the search query term.

Function f – fraction of users who click on a reply.

(Submodularity – it suffices that a user likes just one of the replies.)

What is the best set of k replies to display?

A value query – display a set of replies and observe the fraction of users who clicked on some reply.



Beyond value queries

Example: f is a valuation function of a buyer in a situation such as a combinatorial auction.

Demand query: gives prices to the items, and observes what bundle of items the buyer chooses to buy, or ask the (truthful) buyer which bundle maximizes utility (= value – price).

Very useful for approximating the maximum welfare problem in combinatorial auctions.

In this talk we consider only value queries.

Physical limitations on value queries

Standard models of value query oracles allow the query to be any set $S \subset U$.

This makes perfect sense when queries serve as an abstraction of the process of performing computations on some implicit representation.

This makes less sense for physical queries.

- If a query S involves hiring set S of workers, different queries have different costs.
- External factors might make some queries undesirable. If too few products are displayed in the shop window, passers-by might get the wrong impression that the shop is not yet open for business.

Uniform size queries

One may easily formulate models in which queries have different costs and an algorithm with a limited query budget attempts to approximate max k-submodular.

In this work: a mathematically natural cost model – every feasible solution is a feasible query.

The cost of query S is 1 if $|S|=k$, and infinite otherwise.

In practice one would not expect such an extreme cost model. Some insights acquired for the uniform size cost model apply to other cost models, though this will not be discussed in this talk.

Answers to queries

We think of queries as physical experiments. As such, we expect errors in the replies to queries.

Our algorithmic results hold as long as these errors are sufficiently small (the answer to query S approximates f(S) within $1 \pm \frac{\varepsilon}{k}$).

For simplicity, we ignore these errors in this talk.

Recap: Max k-submodular

Given universe U, submodular function f and a parameter k, output a set S of cardinality k that maximizes f(S).

Access to f : size-k value queries.

Answers to queries: exact.

Algorithms: limited to polynomially many queries and polynomial computation time.

What is the best approximation ratio achievable?

Known results

Cannot do better than $1 - \frac{1}{e}$ (even with implicit representations).

The known algorithms that achieve $1 - \frac{1}{e}$ (e.g., greedy) involve value queries to sets smaller than k, and hence are not applicable.

The interchange heuristic of [Nemhauser, Wolsey, Fisher 1978] can be implemented in the uniform query model and gives approximation ratio of at least $\frac{k}{2k-1+\varepsilon} \cong \frac{1}{2}$.

Our results

Theorem 1: for $\varepsilon \leq \frac{1}{2}$, any algorithm that has probability at least $\frac{1}{2}$ of approximating max-k submodular within a ratio of $\frac{1+\varepsilon}{2}$ must use at least $\Omega(\left(\frac{\varepsilon n}{ek}\right)^{\varepsilon k})$ k-queries.

Theorem 2: there is some $\rho > \frac{1}{2}$ such that for every k, the conditional greedy (randomized) algorithm approximates max-k coverage within a ratio of ρ, and uses at most polynomially many size-k queries.

Our proof shows that $0.50004 < \rho < 0.582$.

Unrelated work

[Radlinski, Kleinberg, Joachims 2008]

A $1 - \frac{1}{e}$ approximation for max-k coverage using size-k queries.

However: a crucial extra assumption.

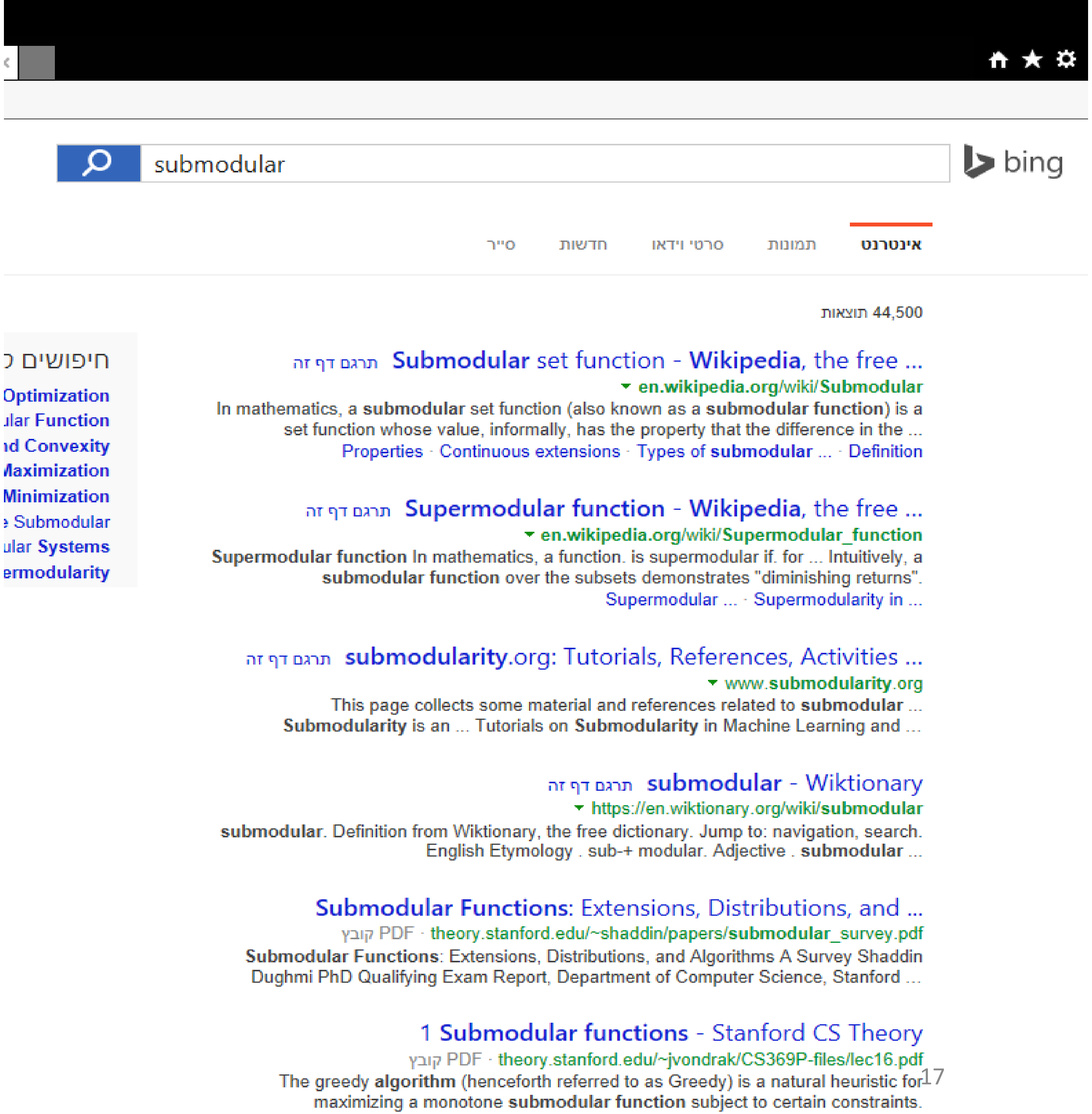
Universe U – Potential replies to the search query term.

Function f – fraction of users who click on a reply.

What is the best set of k replies to display?

A value query – display a set of replies and observe the fraction of users who clicked on some reply.

Assumption: every user scans the replies from top to bottom and clicks on first reply that he/she likes.



[Radlinski, Kleinberg, Joachims 2008]

A $1 - \frac{1}{e}$ approximation for max-k coverage using size-k queries.

Consequence of the assumption: the replies reveal the marginal value of each item in the k-tuple compared to the previous items.

Hence each query implements queries of all sizes 1,2,…,k, and not just a query of size k. Hence can implement the standard greedy algorithm.

In our work we make no such assumptions. Reply to S only reveals the value of f(S) and no further information. (User clicks on arbitrary reply that he/she likes.)

Review: the interchange heuristic (local search) [Nemhauser, Wolsey, Fisher 1978]

Given a set A of size k, move to a set B of size k that differs by one item if $f(B) > f(A)$, or stop if no such set B exists.

A single step can be implemented using $k(n-k)$ size-k queries.

The number of steps is at most $\binom{n}{k}$ (which is not polynomial in k).

A local optimum is at least a $\frac{k}{2k-1}$ approximation

Let S be a local optimum and T an optimum.

By submodularity:

- There is an item $x \in S$ such that $f(S - x) \geq \frac{k-1}{k} f(S)$.
- There is an item $y \in T$ such that

$$f(S - x + y) \geq f(S - x) + \frac{f(T) - f(S - x)}{k} \geq \frac{1}{k} f(T) + \frac{(k-1)^2}{k^2} f(S)$$

By local optimality this last expression is at most $f(S)$.

Algebraic manipulations imply: $f(S) \geq \frac{k}{2k-1} f(T)$.

A polynomial time version

Given a set A of size k, move to a set B of size k that differs by one item if $f(B) > f(A)(1 + \frac{\varepsilon}{k^2})$, or stop if no such set B exists.

A single step can be implemented using $k(n - k)$ size-k queries.

The number of steps is at most $O(\frac{k^2 \log k}{\varepsilon})$ (polynomial in k).

The approximation ratio is at least $\frac{k}{2k-1+\varepsilon}$.

Max 5-coverage example of a bad local optimum

	Local 1	Local 2	Local 3	Local 4	Local 5				
OPT 1									
OPT 2									
OPT 3									
OPT 4									
OPT 5									

Hardness of approximation beyond $\frac{1}{2}$

Two types of items.

k good items. Each of value 1. (rows)

$n-k$ bad items. Each of value $\rho = \frac{1+\varepsilon}{2}$. (virtual columns)

For set T (of any size) with g good items and b bad items

$$f(T) = \min[k, \rho b + g \max\left[\frac{k-\rho b}{k}, \rho\right]]$$

Lemma: f is submodular. (Not a coverage function!)

Lemma: All but a fraction of $(\left(\frac{ek}{\varepsilon n}\right)^{\varepsilon k})$ sets of size k have value $= \rho k$.

Coverage functions – attempted hardness

$\frac{1}{2} < \rho < 1$. Two types of items.

k good items, making the optimal solution T^*.

$n - k$ other items.

$f(T^*) = k$.

Almost all sets of size k have value $= \rho k$.

For what values of ρ does a coverage function f with such properties exist?

Recap: Max k-coverage

Given universe U, coverage function f and a parameter k, output a set S of cardinality k that maximizes f(S).

Access to f : size-k value queries.

Answers to queries: exact.

The items cover elements. However, queries cannot access these elements.

Algorithms: limited to polynomially many queries and polynomial computation time.

A useful abstraction – conditional queries

$$i \in U$$
$$T \subset U - \{i\}$$
$$|T| \leq k - 1$$
$$f_k(i|T) = E_{|S|=k\ ;T \cup \{i\} \subset S} f(S)$$

The conditional query $f_k(i|T)$ is the expected value of a random set S of cardinality k, conditioned on S containing T and i.

Conditional (deterministic) queries can be implemented w.h.p. up to arbitrary precision using polynomially many (random) size-k queries.

(In this talk, omit discussion of the negligible error terms)

The conditional greedy algorithm

1. Initialize T_0 to be the empty set.
2. Repeat for $l = 1$ up to k:
   a. For each $i \in (U - T_{l-1})$ perform the conditional query $f_k(i|T)$.
   Let i^* be the item with highest reply.
   b. Then $T_l = T_{l-1} \cup \{i^*\}$.
3. Output T_k .

The conditional greedy algorithm

Equivalently:

- Randomized algorithm: select a solution at random from the uniform distribution over sets of size k.
- Derandomize it using the method of conditional expectations.

Main theorem

Theorem: There is some $\rho > \frac{1}{2}$ such that for every k, the conditional greedy algorithm approximates max-k coverage within a ratio of ρ, and uses at most polynomially many conditional queries.

Corollary: There is some $\rho > \frac{1}{2}$ such that for every k, the conditional greedy (randomized) algorithm approximates max-k coverage within a ratio of ρ (in expectation), and uses at most polynomially many size-k queries.

Our proof shows that $0.50004 < \rho < 0.582$.

Observations

The underlying randomized algorithm does not give a good approximation ratio. Still, its derandomized version does. (Earlier examples of similar phenomena in other contexts include [Chen, Friesen and Zheng 1999],[Bar-Noy and Lampis 2012].)

For max k-submodular the randomized algorithm does not give an approximation ratio strictly above $\frac{1}{2}$.

Our analysis for max k-coverage must:

- Use properties that are specific to coverage functions.
- Be robust to small errors in replies to the conditional queries.

Notation for proof of main theorem

O – set of elements covered by optimal solution.

l - round number.

$f(T_l)$ - set (sometimes number) of elements covered by T_l .

R_l- set of elements newly covered in round l.

Y_l - expected fraction of elements from $O - f(T_{l-1})$ not covered by $k - l$ random items from $U - T_l$.

Note: $|R_l| \geq \frac{Y_l}{k}|O|$.

Lemma 1

W.l.o.g., in every round $Y_l > \frac{1}{2} - \varepsilon_1$.
Informally: in every round, a random feasible k-set will not cover much more than half of O.

Proof: conditional greedy does at least as well as random. Will cover at least $\left(\frac{1}{2} + \varepsilon_1\right)|O|$.

Lemma 2

W.l.o.g., in almost every round $Y_l < \frac{1}{2} + \varepsilon_2$.
Informally: in most rounds, a random feasible k-set will cover nearly half of O.

Proof: otherwise, in many rounds $R_l \geq (\frac{1}{2} + \varepsilon_2)\frac{|O|}{k}$, implying (roughly) $f(T_k) \geq (\frac{1}{2} + \Omega(\varepsilon_2) - \varepsilon_1)|O|$.

Corollary: taking $\varepsilon_1 \ll \varepsilon_2$ we may assume that $Y_l \cong \frac{1}{2}$ in almost all rounds.

Lemma 3

Recall: we may assume that $Y_l \cong \frac{1}{2}$ in almost all rounds.

W.l.o.g., in almost every round almost all elements of R_l are from O.

Proof: otherwise, $f(T_{0.1k})$ contains at least $\varepsilon_3|O|$ elements not from O. Taking $\varepsilon_3 \gg \varepsilon_2, \varepsilon_1$, the approximate equality $Y_l \cong \frac{1}{2}$ then implies that $f(T_k) \geq \frac{1}{2} + \Omega(\varepsilon_3)$.

Lemma 4

Recall: in almost all rounds $Y_l \cong \frac{1}{2}$ and almost all elements covered by R_l are from O.

W.l.o.g., in a typical round, the elements covered by R_l contributed nearly $\frac{|O|}{2k}$ to Y_l.

Proof: otherwise, the marginal conditional contribution of the item chosen at round l is too small.

A key lemma

In almost every round:

- $Y_l \cong \frac{1}{2}$
- The elements covered by R_l contributed nearly $\frac{|O|}{2k}$ to Y_l.

Y_l should decrease by $\frac{|O|}{2k}$ in almost every round.

Mitigated by an increase: the elements in $O - T_l$ contribute more to Y_{l+1} than to Y_l .

Key lemma: the increase is bounded by roughly $\frac{|O|}{e(k-l)}$.

Recap of proof

To avoid approximation better than $\frac{1}{2}$ must maintain $Y_l \cong \frac{1}{2}$ in almost every round.

The choice R_l decreases Y_l by at least $\frac{|O|}{2k}$ in almost every round.

This is mitigated by an increase of at most $\frac{|O|}{e(k-l)}$ contributed by the elements in $O - T_l$.

In the first $\frac{k}{20}$ rounds there must be a significant downward drift in Y_l.

Instantiating all parameters of the proof shows that $\rho > 0.50004$.

Summary of main results

Polynomially many size-k value queries:

- Max k-submodular can be approximated within $\frac{1}{2}$ but not better.
- The conditional greedy algorithm approximates max k-coverage within a ratio of $0.50004 < \rho < 0.582$.

Open questions

- What is the true approximation ratio of the conditional greedy algorithm for max k-coverage?
- Can max k-coverage be approximated within a ratio of $1 - \frac{1}{e}$ using only size-k queries?

Is the following known?

- Allowing for exponential time algorithms but with only polynomially many value queries (or size-k queries), can max k-coverage be approximated within a ratio better than $1 - \frac{1}{e}$?