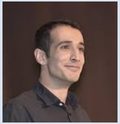
Consistent Submodular Maximization




Paul Dütting — Google

Federico Fusco — **Sapienza**

Silvio Lattanzi — Google

Ashkan Norouzi-Fard — Google

Morteza Zadimoghaddam — Google

ICML 2024

Vienna - Austria - July 24, 2024

Submodularity

Submodularity models diminishing returns:

$$f(S + x) - f(S) \geq f(T + x) - f(T), \quad \forall S \subseteq T \subseteq V, \ \forall x \in V \setminus T$$

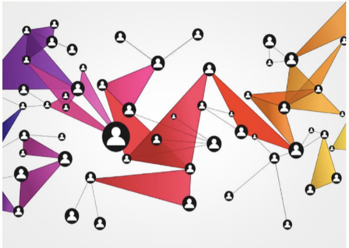


Figure 1: Influence Maximization

Figure 2: Data Summarization

Consistent submodular maximization

- Elements arrive online $e_1, e_2, \ldots$
- $V_t = \{e_1, \ldots, e_t\} \subseteq V$ is the set of first t elements
- OPT_t is the best subset of k elements in V_t
- **Goal**: maintain a solution S_t, with $|S_t| \leq k$ such that
  1. (Approximation) $f(\mathrm{OPT}_t) \leq \alpha f(S_t)$
  2. (C-Consistency) $|S_t \setminus S_{t-1}| \leq C \in O(1)$

Our Result

- Previous Results:
  - Recomputing the solution *from scratch* is a $e/e-1$ approx. but is $\Omega(k)$-consistent.
  - Swapping by Chakrabarti & Kale (2015) is a consistent 4-approximation.
- Our *first* algorithm, Encompassing-Set maintains a $(3.147 + O(1/k))$-approximate solution and is 1-consistent.
- Our *second* algorithm, Chasing-Local-Opt takes in input a precision ε and maintains a $(2.619 + \varepsilon)$-approximation algorithm that is $\tilde{O}(1/\varepsilon)$-consistent
- We prove that *no deterministic* algorithm can maintain an approximation better than 2, while enforcing consistency

Encompassing-Set

Algorithm 1 Encompassing-Set

Environment: Stream V, function f, cardinality k
Threshold parameter $\beta \leftarrow 1.14$
$B_0 \leftarrow \emptyset$, $S_0 \leftarrow \emptyset$, and $t \leftarrow 1$
for e_t new element arriving **do**
  if $f(e_t \mid B_{t-1}) \geq \frac{\beta}{k} f(B_{t-1})$ **then**
    $B_t \leftarrow B_{t-1} + e_t$
    $S_t \leftarrow S_{t-1} + e_t$
    if $|S_t| = k + 1$ **then**
      remove from S_t the element e_s with smallest s
    $t \leftarrow t + 1$

- A benchmark set B_t is used to decide whether to add or discard any new element e_t
- The solution S_t contains the *last* k elements added to B_t

- The benchmark has *good* value compared to OPT_t
- The elements in $B_t \setminus S_t$ are *geometrically* smaller than the ones still in S_t

Chasing-Local-Opt

Algorithm 2 Min-Swap(S, x)

Input: Set S and element x
Environment: Function f and cardinality k
if $|S| < k$, **then return** $S + x$
Let $r \in S$ be any element s.t. $f(r \mid S - r) \leq f(S)/k$
return $S - r + x$

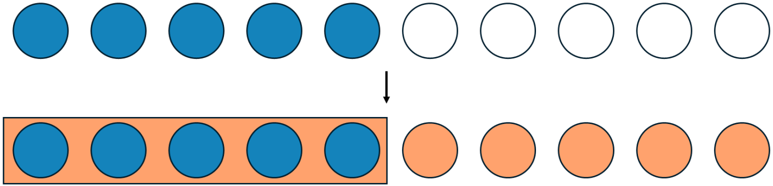
Algorithm 3 Chasing-Local-Opt

Input: Precision parameter ε
Environment: Stream V, function f, cardinality k
$\phi \leftarrow \frac{\sqrt{5}+1}{2}$, $N \leftarrow \lceil \frac{1}{\varepsilon} \log_\phi \frac{12}{\varepsilon} \rceil$
$S_0 \leftarrow \emptyset$ and $t \leftarrow 1$
for e_t new element arriving **do**
  if $f(e_t \mid S_{t-1}) \geq \frac{\phi}{k} f(S_{t-1})$ **then**
    $S_t \leftarrow$ Min-Swap(S_{t-1}, e_t)
  for $i = 1, \ldots, N$ **do**
    if $\exists\, x \in V_t$ such that $f(x \mid S_t) \geq \frac{\phi}{k} f(S_t)$ **then**
      $S_t \leftarrow$ Min-Swap(S_t, x)
  $t \leftarrow t + 1$

- Chasing-Local-Opt updates the solution S_t via local improvements.
- If S_t is an approx. local optimum then it is a good approximation of OPT_t, as no element $x \in \mathrm{OPT}_t$ verifies $f(x \mid S) \geq \phi/k \cdot f(S)$
- If this is not the case, then if means that *many local swaps* have happened, so that the value of the solution has increased a lot

Impossibility Result

Consider a covering instance on an universe of n elements, and $k = n/2$.



1. n singletons arrive, and the *algorithm* selects half of them
2. A subset covering the selected elements arrives, so that the optimal solution has value n, as opposed to the *algorithm* that cannot significantly improve over $n/2$.

Future Directions

- Would randomization actually help?
- Can we maintain consistency in the *fully dynamic* setting?
- Can we tackle more *complex* constraints?

Thank you!