INSTRUCTOR: VOLKAN CEVHER SCRIBER: YA-PING, ILIJA

# ROBUST SUBMODULAR MAXIMIZATION

In many applications, we want to select observations which are robust against a number of possible objective functions. Examples include sensor placement for outbreak detection, blocking mechanisms to suppress the spread of misinformation in networks, etc. We study the submodular Saturation algorithm, a simple and efficient algorithm with strong theoretical approximation guarantees for cases where the possible objective functions exhibit submodularity, an intuitive diminishing returns property.

We start this lecture by first defining the robust submodular function maximization problem. Next, we describe a few real-world problems that can be formulated as the robust submodular maximization. In the subsequent section, we discuss the hardness of this problem. In Section 4, we revisit the MinCover problem and the Greedy Partial Cover algorithm. Finally, we describe the Saturate algorithm and discuss its theoretical guarantees.

The outline that summerizes this lecture:

1. Robust submodular function maximization
2. Hardness results and performance of the greedy algorithm
3. MinCover problem and the Greedy Partial Cover algorithm
4. Algorithm overview
5. The Saturate algorithm

## 1 Robust submodular maximization

Let us start by recalling the definition of submodular functions.

**Definition 1.** *A function $f : 2^V \to \mathbb{R}$ on a ground set $V$ is said to be submodular if for all $S \subseteq T \subseteq V$ and any $e \in V \backslash T$, it holds $\Delta(e|S) \geq \Delta(e|T)$, where $\Delta(e|S) = f(S \cup \{e\}) - f(S)$.*

We also call $f$ monotone, if for all $S \subseteq T$, we have that $f(S) \leq f(T)$.

**Problem 1** (Robust submodular maximization problem - RSFMax)**.** *Given a collection of normalized monotonic submodular functions $f_1, ..., f_m$, find a set $S \subseteq V$, which is robust against the worst possible objective, $\min_i f_i$ ($i \in \{1, ..., m\}$):*

$$\max_{S \subseteq V} \min_i f_i(S), \qquad \textit{subject to} \quad |S| \leq k$$

In what follows we describe a few real-world problems that can be captured by the previous problem formulation.

## 2 Examples

### 2.1 Submodular maximization in learning-based CS.

First, let us recall the learning based compressive sensing problem and average energy criterion.

**Problem 2** (LB-CS: Problem statement)**.** *Given a set of $m$ training signals $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbf{C}^p$, find an index set $\Omega$ of a given cardinality $n$ such that a related test signal $\boldsymbol{x}$ can reliably be recovered given the subsampled measurement vector $\mathbf{b} = \mathbf{P}_\Omega \mathbf{\Psi} \boldsymbol{x}$.*

**Problem 3** (Average energy criterion)**.**

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \Omega} |\langle \psi_i, \boldsymbol{x}_j \rangle|^2$$

This is a cardinality constrained modular maximization problem. Alternatively, we can also formalize another criterion in the following way:

**Problem 4** (Worst-case energy criterion)**.**

$$\hat{\Omega} = \underset{\Omega : |\Omega| = n}{\arg\max} \min_{j=1,\dots,m} \sum_{i \in \Omega} |\langle \psi_i, \boldsymbol{x}_j \rangle|^2.$$

***This is an instance of the robust modular maximization problem.***

In **Problem 4**, we care more about the worst case (minimum energy) scenario, whereas in **Problem 3** we are concerned with average case. Compared to the average criterion, the worst-case criterion may be preferable in some cases, but it tends to be less robust to "outliers".

**Interpretation 1: Linear decoding performance.** Here, we digress a little bit to answer a very important question. Capturing energy sounds like a reasonable criterion, but does it actually correspond to good recovery performance? The answer is yes for a particular choice of decoder.

**Definition 2** (Linear decoder)**.** *We consider a linear decoder that expands* $\mathbf{b}$ *to a p-dimensional vector by placing zeros in the entries corresponding to* $\Omega^c$*, and then applies the adjoint* $\mathbf{\Psi}^* = \mathbf{\Psi}^{-1}$*:*

$$\hat{\boldsymbol{x}} = \mathbf{\Psi}^* P_{\Omega^.}^T$$

**Theorem 2.1.** *The* $\ell_2$ *estimation error of the above decoder is*

$$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 = \|\boldsymbol{x}\|_2^2 - \|P_\Omega \mathbf{\Psi} \boldsymbol{x}\|_2^2.$$

*Proof.*

$$\begin{aligned}
\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 &= \|\mathbf{x} - \mathbf{\Psi}^* P_\Omega^T P_\Omega \mathbf{\Psi} x\|_2^2 &(1)\\
&= \|\mathbf{\Psi}\mathbf{x} - P_\Omega^T P_\Omega \mathbf{\Psi} x\|_2^2 &(2)\\
&= \|P_{\Omega^c}^T P_{\Omega^c} \mathbf{\Psi} x\|_2^2 &(3)\\
&= \|P_{\Omega^c} \mathbf{\Psi} x\|_2^2 &(4)
\end{aligned}$$

where (2) follows since $\mathbf{\Psi}$ is an orthonormal basis matrix, (3) follows since $P_\Omega^T P_\Omega + P_{\Omega^c}^T P_{\Omega^c} = I$, and (3) follows since a multiplication by $P_{\Omega^c}^T$ simply produces additional rows that are equal to zero. The theorem then follows since

$$\|\mathbf{x}\|_2^2 = \|P_\Omega \mathbf{\Psi} \mathbf{x}\|_2^2 + \|P_{\Omega^c} \mathbf{\Psi} x\|_2^2. \tag{5}$$

□

The previous theorem shows that maximizing the captured energy in the worst case, amounts to minimizing the error of the linear decoder.

**Interpretation 2: Subsampling pattern providing the best restricted isometry property (RIP) constant.**

We have $\|P_\Omega \mathbf{\Psi} \mathbf{x}_j\|_2 \le \|\mathbf{x}_j\|_2$ ($\mathbf{\Psi}$ is an orthonormal basis matrix). Thus, defining $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_m]$ and $\mathbf{V} := \mathbf{\Psi}\mathbf{X}$, we can equivalently write as

$$\hat{\Omega} = \underset{\Omega : |\Omega| = n}{\arg\min} \|\mathbf{1} - \mathrm{diag}(\mathbf{V}^T P_\Omega^T P_\Omega \mathbf{V})\|_\infty, \tag{6}$$

where $\mathbf{1}$ is the vector of $m$ ones, and $\mathrm{diag}(\cdot)$ forms a vector by taking the diagonal entries of a matrix.

In this form, the optimization problem can also be interpreted as finding the subsampling pattern providing the best restricted isometry property (RIP) constant [5] with respect to the training [1, 2].

**Generalization bounds.** Capturing as much of the signal energy as possible on the training signals $\mathbf{x}_j$ corresponds to minimizing the $\ell_2$-norm error of the linear decoder. The following theorem provides the answer under which conditions the same will be true for some new signal $\mathbf{x}$.

**Theorem 2.2** (Deterministic generalization bound for $f = f_{\min}$ [3])**.** *Fix* $\delta > 0$ *and* $\epsilon > 0$*, and suppose that for a set of training signals* $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ *with* $\|\boldsymbol{x}_j\|_2 = 1$*, we have a sampling set* $\Omega$ *such that*

$$\min_{j=1,\dots,m} \|P_\Omega \mathbf{\Psi} \boldsymbol{x}_j\|_2^2 \ge 1 - \delta. \tag{7}$$

*Then for any signal* $\boldsymbol{x}$ *with* $\|\boldsymbol{x}\|_2 = 1$ *such that* $\|P_{\Omega^c} \mathbf{\Psi}(\boldsymbol{x} - \boldsymbol{x}_j)\|_2^2 \le \epsilon$ *for some* $j \in \{1, \dots, m\}$*, we have*

$$\|P_\Omega \mathbf{\Psi} \boldsymbol{x}\|_2^2 \ge 1 - (\sqrt{\delta} + \sqrt{\epsilon})^2. \tag{8}$$

*Proof.* It follows from (5) and (7) that $\|P_{\Omega^c} \mathbf{\Psi} \mathbf{x}_j\|_2^2 \le \delta$ for all $j$. Hence, letting $j$ be an index such that $\|P_{\Omega^c} \mathbf{\Psi}(\mathbf{x} - \mathbf{x}_j)\|_2^2 \le \epsilon$, we obtain from the triangle inequality that

$$\|P_{\Omega^c} \mathbf{\Psi} \mathbf{x}\|_2 \le \|P_{\Omega^c} \mathbf{\Psi} \mathbf{x}_j\|_2 + \|P_{\Omega^c} \mathbf{\Psi}(\mathbf{x} - \mathbf{x}_j)\|_2 \le \sqrt{\delta} + \sqrt{\epsilon}. \tag{9}$$

Taking square and applying (5), we obtain (8). □

Therefore, if a new sample is close to any of the training examples, and we have a guarantee for the its energy captured by the sampling set $\Omega$.

### 2.2 Protection of networks against cascading phenomena

More real-world problems that can be formulated as the robust submodular maximization problems are:

- Sensor placement for outbreak detection [6]
- Protection of networks against cascading phenomena [4]

In sensor placement, the NIMS (Networked InfoMechanical System) robot shown in Figure 2.2 is deployed in order to estimate the pH values across a horizontal transect of a river. The goal is to deploy these robots in such a way that they quickly discover the spread of a contagion. The deployment strategy should be robust w.r.t. to the worst possible contagion outcome. Next, we explain the second real-world problem in more details.

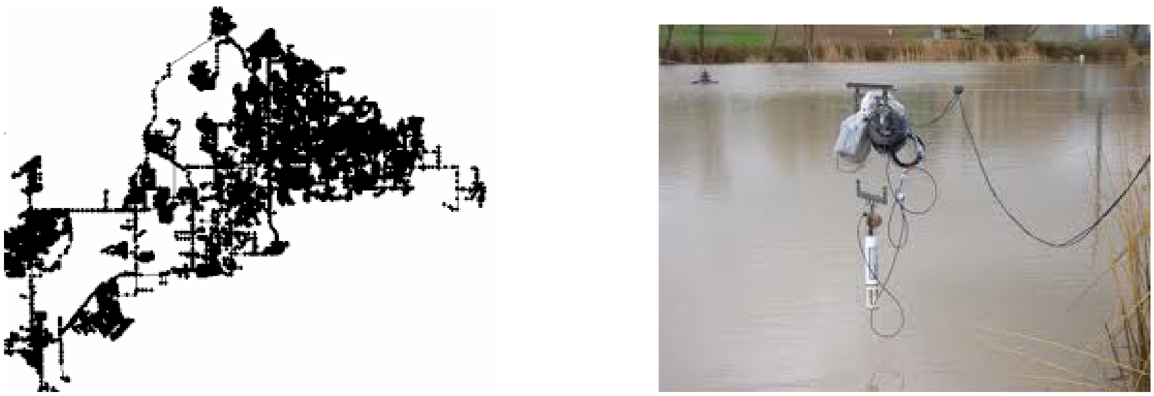


Figure 1: Illustration of the municipal water distribution network and Deployment of the Networked Infomechanical System

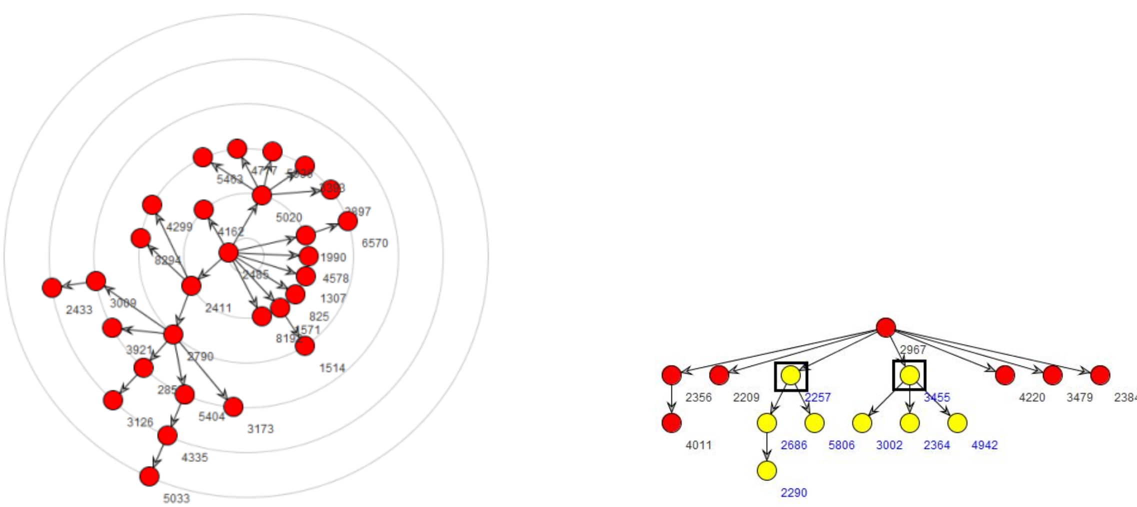


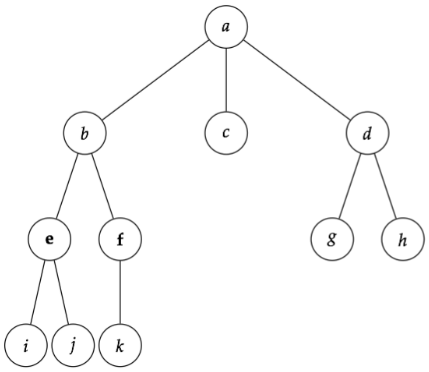

Figure 2: Examples of cascades

In Figure 2.2 , we show an example of a cascade that can spread in the network $G(V, E)$. Usually, cascades have tree structures (different cascade models are explained in [4]). We assume the following protection mechanism: if a cascade $c(V_c, E_c)$ contains a *blocking node* $b$, all descendant nodes of node $b$ in $c$ become protected (yellow nodes in Figure 2.2).

We use $F_c$ to denote the number of protected nodes in the cascade $c$,

$$F_c(B) := \left| \bigcup_{b \in B} \text{descendants}_c(b) \right|. \tag{10}$$

The number of protected nodes $F_c$ is a coverage function, and the coverage functions are submodular by definition. The diminishing returns behavior of this function is illustrated in Figure 3.




$A = \{f\}$

$B = \{e, f\}$

$F(A) = 2$

$F(B) = 5$

| s | $F(A \cup \{s\}) - F(A)$ | $F(B \cup \{s\}) - F(B)$ |
|---|---|---|
| a | **9** (a, b, c, d, e, g, h, i, j) | 6 (a, b, c, d, g, h) |
| b | **4** (b, e, i, j) | 1 (b) |
| c | 1 (c) | 1 (c) |
| d | 3 (d, g, h) | 3 (d, g, h) |
| g | 1 (g) | 1 (g) |
| h | 1 (h) | 1 (h) |
| i | **1** (i) | 0 |
| j | **1** (j) | 0 |
| k | 0 | 0 |

Figure 3: Diminishing returns

Next, we define our objective function, the number of protected nodes in the network.

**Definition 3** (The number of protected nodes)**.** *For a given cascade $c(V_c, E_c)$ and a set of blocking nodes $B$, let $S_c$ denote the number of protected nodes in the network,*

$$S_c(B) = F_c(B) + \lambda_c, \tag{11}$$

*where $\lambda_c$ is the difference between the size of the network and the size of the cascade $c$, i.e., $\lambda_c = |V| - |V_c|$.*

Finally, we can formulate the robust protection of networks problem which is an instance of the robust submodular maximization problem.

**Problem 5** (Robust protection of networks)**.** *Given a directed network $G(V, E)$ and an arbitrary set of cascades $C$, $|C| \leq m$ that can possibly spread in $G$, find a set of nodes $B$ to block so that*

$$\max_{B \subseteq V} \min_{c} S_c(B) \quad s.t. \quad |B| \leq k, \tag{12}$$

*i.e., the protection against the worst-possible cascade outcome is maximized.*

## 3 Hardness of the RSFMax problem

In Problem 1, $f_i$ are all submodular, but $f_{wc}(S) := \min_i f_i(S)$ is generally not submodular. Hence, the simple greedy algorithm (which performs near-optimally in the single-criterion setting) can perform arbitrarily badly. Example of this is given in Table 3. Given, $V = \{s_1, s_2, t_1, t_2\}$ and $k = 2$, the greedy algorithm maximizing $f_{wc}(S) = \min\{f_1(S), f_2(S)\}$ would choose $\{t_1, t_2\}$ obtaining a score of 2. The optimal solution for $k = 2$ is $\{s_1, s_2\}$, with a score of $n$. As, $n \to \infty$, the greedy algorithm performs arbitrarily worse!

| **S** | $\mathbf{f_1(S)}$ | $\mathbf{f_2(S)}$ | $\min_\mathbf{i} \mathbf{f_i(S)}$ |
|---|---|---|---|
| $\emptyset$ | 0 | 0 | 0 |
| $\{s_1\}$ | $n$ | 0 | 0 |
| $\{s_2\}$ | 0 | $n$ | 0 |
| $\{t_1\}$ | 1 | 1 | 1 |
| $\{t_2\}$ | 1 | 1 | 1 |
| $\{s_1, s_2\}$ | $n$ | $n$ | $n$ |
| $\{s_1, t_1\}$ | $n+1$ | 1 | 1 |
| $\{s_1, t_2\}$ | $n+1$ | 1 | 1 |
| $\{s_2, t_1\}$ | 1 | $n+1$ | 1 |
| $\{s_2, t_2\}$ | 1 | $n+1$ | 1 |
| $\{t_1, t_2\}$ | 2 | 2 | 2 |

Table 1: Functions $f_1$ and $f_2$ are used in counterexample.

The following theorem states that solving the RSFMax problem approximately is NP-hard.

**Theorem 3.1** (Hardness of Approximate Solution [6])**.** *If there exists a positive function $\gamma(\cdot) > 0$ and an algorithm that, for all n and k, in time polynomial in the size of the problem instance n, is guaranteed to find a set $S'$ of size k such that*

$$\min_i f_i(S') \geq \gamma(n) \max_{|S| \leq k} \min_i f_i(S),$$

*then $P = NP$.*

In other words: there cannot exist any polynomial time approximation algorithm for the *RSFMax* problem (unless $P = NP$).

In the next section, we explain the minimum submodular set cover problem and the algorithm used to solve it, as this is going to be one of the main components for solving the relaxed version of the RSFMax problem.

## 4 Minimum submodular set cover (MinCover)

**Problem 6** (MinCover)**.** *For any given c solve:*

$$S_c = \arg\min_{S \subseteq V} |S| \quad \text{subject to} \quad f_i(S) \geq c \text{ for } 1 \leq i \leq m,$$

*i.e., find the smallest set $S$ with $f_i(S) \geq c$ for all i.*

In the case of a single criterion, i.e., when $m = 1$, we know that the following theorem holds.

**Theorem 4.1** ([7])**.** *Given a submodular integer-valued function $f$ and a fixed $c \in \mathbb{Z}, c \leq f(V)$. Let $S_l$ be the greedy solution and let $\ell$ be the smallest integer such that $f(S_l) \geq c$. Then*

$$\ell \leq \left(1 + \ln \max_{v \in V} f(\{v\})\right) k^\star$$

We note that there are non-integer variants of this theorem [7].

Next, we show a simple trick that allows us to use the result of the previous theorem in the case when we have multiple objective functions, i.e. $m > 1$.

**Definition 4.** *We define the following functions:*

$$\hat{f}_{i,c}(S) := \min\{f_i(S), c\} \qquad \bar{f}_c(S) := \frac{1}{m} \sum_i \hat{f}_{i,c}(S) \tag{13}$$

The previous transformations preserve submodularity. Now, we can rewrite the *MinCover* problem as:

$$S = \arg\min_{S \subseteq V} |S| \quad \text{subject to} \quad \bar{f}_c(S) = c.$$

Notice, that in this formulation, we have a single objective function. We can solve this problem by using the GPC (Greedy partial cover) algorithm (Algorithm 1).

**Algorithm 1** Greedy Partial Cover

**GPC**($\bar{f}_c, c$)**:**

1: $S \leftarrow \emptyset$
2: **while** $\bar{f}_c(S) < c$ **do**
3: $\quad \Delta_j \leftarrow \bar{f}_c(S \cup \{j\}) - \bar{f}_c(S)$
4: $\quad S \leftarrow S \cup \arg\max_j \Delta_j$
5: **return** $S$

Finally, we can use the result of Theorem 4.1 to show the following:

**Theorem 4.2** (Approximation achieved by the GPC algorithm [6])**.** *Given integer monotonic submodular functions $f_1, ..., f_m$ and a constant c, GPC with input $\bar{f}_c$ finds a set $S_l$ such that $f_i(S_l) \geq c$ for all i, and $|S_l| \leq \alpha k^*$, where $k^*$ is the size of the optimal solution to Problem* **??** *, and*

$$\alpha = 1 + \ln\left(\max_{v \in V} \sum_i f_i(\{v\})\right)$$

An important observation is that $\alpha$ does not depend on $c$, hence we can compute alpha for any $c$. We will use the GPC algorithm as the subroutine in the submodular saturation algorithm.

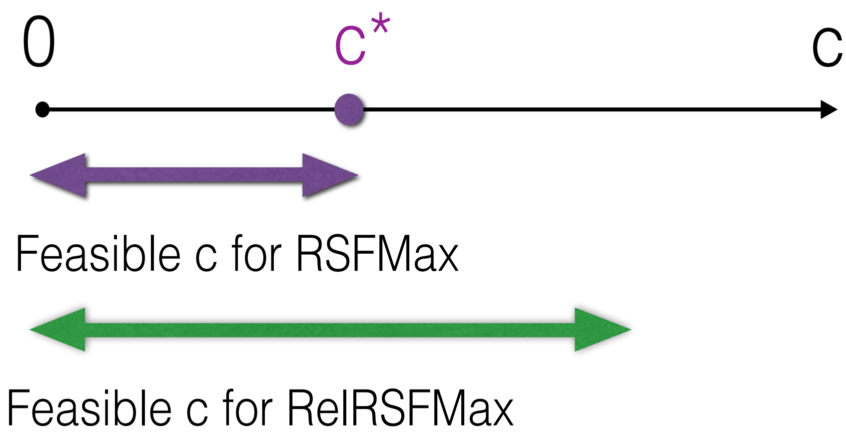



Figure 4: Illustration of feasible regions for the RSFMax and RelRSFMax problems. $c^*$ is the optimal solution to the RSFMax problem.

## 5 Constraint relaxation

We saw in Section 3 that solving the RSFMax problem approximately is NP-hard. Next, we define its relaxed version.

**Problem 7** (Relaxed RSFMax)**.** RelRSFMax, *the relaxed version of the* RSFMax *problem:*

$$\max_{c,S} c, \qquad \text{subject to} \quad f_i(S) \geq c \text{ for } 1 \leq i \leq m \text{ and } |S| \leq \alpha k$$

*Here,* $\alpha \geq 1$ *is a parameter relaxing the constraint on* $|S|$*. When* $\alpha = 1$, RelRSFMax = RSFMax.

In Figure 5 , we show the illustration of the feasible regions for RSFMax and its relaxed version.

The overall idea to solve the previous problem is as follows:

- set $\alpha = 1 + \ln\left(\max_{v \in V} \sum_i f_i(\{v\})\right)$ in *RelRSFMax*
- for a given $c$ solve *MinCover* problem approximately by using the GPC algorithm
- if $S_c \leq \alpha k$ then both $S_c$ and $c$ are feasible solution to *RelRSFMax* problem
- use binary search to find the solution $S_c \leq \alpha k$ with the maximum feasible $c$

In the next section, we introduce the Saturate algorithm that is based on the previous description.

## 6 The Saturate algorithm

The pseudocode of the algorithm is shown in Algorithm 6. The main idea of Saturate is as follows (see Figure 5 for the illustration):

- Maintain a lower bound ($c_{\min}$) for *RelRSFMax* and an upper bound for *RSFMax* ($c_{\max}$); Initialize $[c_{\min}, c_{\max}] = [0, \min_i f_i(V)]$
- Successively improve the upper and lower bounds using a binary search procedure
- Invoke the GPC algorithm with $c = (c_{\max} + c_{\min})/2$:
  - $|S_c| > \alpha k$ implies that $c > c^*$, hence $c$ is an upper bound to the RSFMax problem; It is safe to set $c_{\max} = c$
  - $|S_c| \leq \alpha k$ implies that $S_c$ is a feasible solution to the RelRSFMax problem; $S_c$ is then kept as best current solution and we can set $c_{\min} = c$
- Upon convergence, we are thus guaranteed a feasible solution to *RelRSFMax* $(c', S')$ such that:

$$c' \geq c^* \quad \text{and} \quad |S'| \leq \alpha k$$

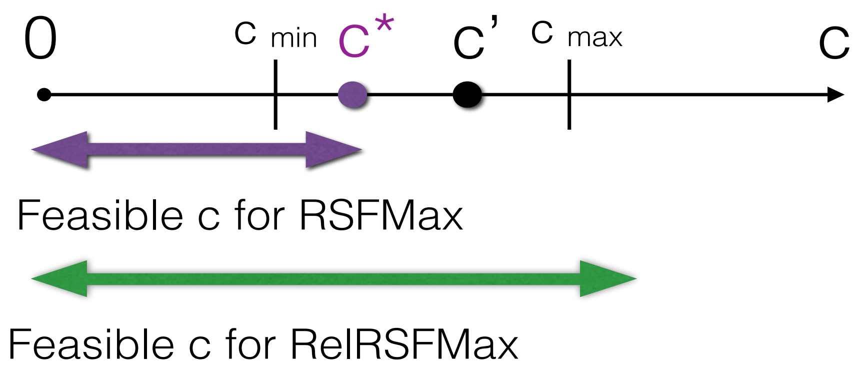



Figure 5: $[c_{\min}, c_{\max}]$ is the search interval during some iteration of Saturate. $c^*$ is the optimal solution to the RSFMax problem, and $c'$ is the solution that will eventually be returned by Saturate.

**Algorithm 2** The Saturate algortihm

**Saturate** $(f_1, \cdots, f_m, k, \alpha)$**:**

1: $c_{\min} \leftarrow 0; c_{\max} \leftarrow \min_j f_j(V); S_{\text{best}} \leftarrow \emptyset$
2: **while** $(c_{\max} - c_{\min}) > 1/m$ **do**
3: $\quad c \leftarrow (c_{\min} + c_{\max})/2$
4: $\quad \bar{f}_c(S) \leftarrow \frac{1}{m}\sum_{j=1}^{m} \min\{f_j(S), c\}$
5: $\quad \hat{S} \leftarrow \text{GPC}(\bar{f}_c, c)$
6: $\quad$ **if** $|\hat{S}| > \alpha k$ **then**
7: $\qquad c_{\max} \leftarrow c$
8: $\quad$ **else**
9: $\qquad c_{\min} \leftarrow c; S_{\text{best}} \leftarrow \hat{S}$
10: **return** $S_{\text{best}}$

**Theorem 6.1** (Approximation achieved by the Saturate algorithm [6])**.** *For any integer $k$,* Saturate *finds a solution $S_{best}$ such that*

$$\min_i f_i(S_{best}) \geq \max_{|S|\leq k} \min_i f_i(S) \quad \textit{and} \quad |S_{best}| \leq \alpha k$$

*for* $\alpha = 1 + \ln\left(\max_{v\in V} \sum_i f_i(\{v\})\right)$.

*Proof.* Let $S^*$ denote an optimal solution to the RSFMax problem. At every iteration of the saturation algorithm it holds that (due to the GPC Theorem)

$$\min_i f_i(S^*) \leq c_{\max},$$

and

$$\min_i f_i(S_{\text{best}}) \geq c_{\min} \quad \text{and} \quad |S_{\text{best}}| \leq \alpha k.$$

Since $f_i$ are integer functions, if $c_{\max} - c_{\min} < \frac{1}{m}$ then it must hold that

$$\min_i f_i(S_{\text{best}}) \geq \min_i f_i(S^*)$$

□

This theorem says that we can achieve the goal of simultaneously optimizing over several submodular functions, provided that we relax the constraint by a factor of $\alpha = 1 + \ln\left(\max_{v\in V} \sum_i f_i(\{v\})\right)$.

# 7 Appendix

Summary of submodular optimization problems covered:

| Lecture | Problem | Algorithm | Approximation | Hardness |
|---|---|---|---|---|
| 2 | Unconstrained SFMax | Greedy | $1/2$ | $(1+\epsilon)1/2$ |
| 2 | Cardinality constrained monotone SFMax | Greedy | $1-1/e$ | $1-1/e$ |
| 2 | Unconstrained MFMax/MFMin | Pick positive weights | 1 | 1 |
| 2 | Cardinality constrained MFMax/MFMin | Sorting | 1 | 1 |
| 2 | Unconstrained SFMin | Convex methods | 1 | 1 |
| 2 | TU constrained MFMax/MFMin | Linear programming | 1 | 1 |
| 4 | Robust monotone SFMax | Saturate | Bicriterion: $(1,\alpha)$ | Bicriterion: $(1,(1-\epsilon)\alpha)$ |

$$\text{where } \alpha = 1 + \ln\Big(\max_{v\in V} \textstyle\sum_i f_i(\{v\})\Big)$$

- SFMax: Submodular function maximization
- SFMin: Submodular function minimization
- MFMax/MFMin: Modular function maximization/minimization

## References


[1] Bubacarr Bah, Steffen Becker, Volkan Cevher, and Baron Gozcu. Metric learning with rank and sparsity constraints. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 21–25. Ieee, 2014.

[2] Bubacarr Bah, Ali Sadeghian, and Volkan Cevher. Energy-aware adaptive bi-lipschitz embeddings. *arXiv preprint arXiv:1307.3457*, 2013.

[3] Luca Baldassarre, Yen-Huan Li, Jonathan Scarlett, Baran Gözcü, Ilija Bogunovic, and Volkan Cevher. Learning-based compressive subsampling. *arXiv preprint arXiv:1510.06188*, 2015.

[4] Ilija Bogunovic. *Robust protection of networks against cascading phenomena*. PhD thesis, Master Thesis ETH Zürich, 2012, 2012.

[5] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser, 2013.

[6] Andreas Krause, H Brendan McMahon, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. 2008.

[7] Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.