Advanced Topics in Data Sciences

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 4: Robust Submodular Maximization

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-731 (Spring 2016)

License Information for Mathematics of Data Slides

Outline

- 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

Recommended Reading

- *Robust submodular observation selection*, Krause, McMahan, Guestrin and Golovin, 2008
- *An analysis of the greedy algorithm for the submodular set covering problem, Wolsey, 1982*
- *Learning-based compressive subsampling, Baldassarre, Li, Scarlett, Gözcü, Bogunovic, and Cevher, 2015*
- *Submodular function maximization, Krause and Golovin, 2012*

Robust submodular maximization

Problem (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*, $\min_i f_i$ *(*$i \in \{1, ..., m\}$*):*

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

Example: Submodular maximization in learning-based CS

LB-CS: Problem statement

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

Average energy criterion

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

This is a cardinality constrained modular maximization problem.

Example: Submodular maximization in learning-based CS

LB-CS: Problem statement

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

Average energy criterion

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

This is a cardinality constrained modular maximization problem.

What about worst-case?

Worst-case energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \min_{j=1,\ldots,m} \sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2.$$

This is an instance of the robust modular maximization problem.

Note: The worst-case criterion may be preferable in some cases, but it tends to be less robust to "outliers", e.g., compared to the average criterion.

Interpretation 1: Linear decoding performance

Capturing energy sounds like a reasonable criterion, but does it actually correspond to good recovery performance?

Linear decoder

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

$$\hat{\mathbf{x}} = \mathbf{\Psi}^* \mathbf{P}_\Omega^T \mathbf{b}.$$

Interpretation 1: Linear decoding performance

Capturing energy sounds like a reasonable criterion, but does it actually correspond to good recovery performance?

Linear decoder

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

$$\hat{\mathbf{x}} = \mathbf{\Psi}^* \mathbf{P}_\Omega^T \mathbf{b}.$$

Theorem

The ℓ_2 estimation error of the above decoder is

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

Worst-case energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \min_{j=1,\ldots,m} \|\mathbf{P}_\Omega \mathbf{\Psi} \mathbf{x}_j\|_2^2$$

Note: 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

Worst-case energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \min_{j=1,\ldots,m} \|\mathbf{P}_\Omega \mathbf{\Psi} \mathbf{x}_j\|_2^2$$

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

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

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

Note: In this form, the optimization problem can also be interpreted as finding the subsampling pattern providing the best restricted isometry property (RIP) constant [4] with respect to the training [1, 6].

Generalization bounds

Capturing as much of the signal energy as possible on the training signals $\mathbf{x}_j$ corresponds to minimizing the ℓ_2-norm error of the linear decoder.

Will the same be true on a new signal $\mathbf{x}$?

Theorem (Deterministic generalization bound for $f = f_{\min}$ [2])

Fix $\delta > 0$ and $\epsilon > 0$, and suppose that for a set of training signals $\mathbf{x}_1, \ldots, \mathbf{x}_m$ with $\|\mathbf{x}_j\|_2 = 1$, we have a sampling set Ω such that

$$\min_{j=1,\ldots,m} \|\mathbf{P}_\Omega \boldsymbol{\Psi} \mathbf{x}_j\|_2^2 \geq 1 - \delta. \tag{2}$$

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

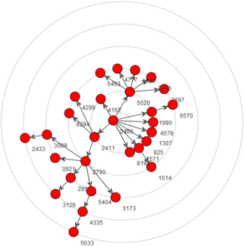
$$\|\mathbf{P}_\Omega \boldsymbol{\Psi} \mathbf{x}\|_2^2 \geq 1 - \left(\sqrt{\delta} + \sqrt{\epsilon}\right)^2. \tag{3}$$

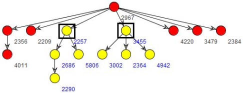
Exercise: Prove the previous theorem.

More examples

- Sensor placement for outbreak detection [5]

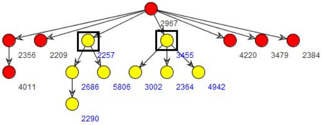
- Protection of networks against cascading phenomena [3]





Example: Protection of networks against cascading phenomena

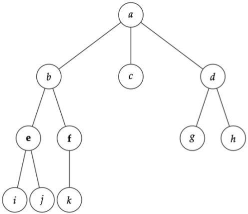
Protection mechanism: if a cascade $c(V_c, E_c)$ contains a *blocking node* b, all descendant nodes of node b in c become protected.



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

$$F_c(B) := \left| \bigcup_{b \in B} \text{descendants}_c(b) \right|. \tag{4}$$

Diminishing returns:



$A = \{f\}$

$B = \{e, f\}$

$F(A) = 2$

$F(B) = 5$

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

Example: Protection of networks against cascading phenomena

Definition (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,

$$S_c(B) = F_c(B) + \lambda_c, \tag{5}$$

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

Problem (Robust protection of networks)

Given a directed network $G(V, E)$ and an arbitrary set of cascades $\mathcal{C}$, $|\mathcal{C}| \leq m$ that can possibly spread in G, find a set of nodes B to block so that

$$\max_{B \subseteq V} \min_{c} S_c(B) \quad \textit{s.t.} \quad |B| \leq k, \tag{6}$$

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

Outline

- 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

Hardness of the RSFMax problem

Problem (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, $\min_i f_i$ ($i \in \{1, ..., m\}$):

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

- **Note:** f_i are all submodular, but $f_{wc}(S) := \min_i f_i(S)$ is generally not submodular
- The simple greedy algorithm (which performs near-optimally in the single-criterion setting) can perform arbitrarily badly

The simple greedy algorithm

$\mathbf{S}$	$\mathbf{f_1(S)}$	$\mathbf{f_2(S)}$	$\min_\mathbf{i} \mathbf{f_i(S)}$
$\emptyset$	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

- Given, $V = \{s_1, s_2, t_1, t_2\}$ and $k = 2$, the greedy algorithm maximizing $f_{wc}(S)$ would choose obtaining a score of

The simple greedy algorithm

$\mathbf{S}$	$\mathbf{f_1(S)}$	$\mathbf{f_2(S)}$	$\min_{\mathbf{i}} \mathbf{f_i(S)}$
$\emptyset$	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

- Given, $V = \{s_1, s_2, t_1, t_2\}$ and $k = 2$, the greedy algorithm maximizing would choose obtaining a score of

The simple greedy algorithm

$\mathbf{S}$	$\mathbf{f_1(S)}$	$\mathbf{f_2(S)}$	$\min_\mathbf{i} \mathbf{f_i(S)}$
$\emptyset$	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

- 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

Solving the RSFMax problem approximately is NP-hard

Theorem (Hardness of Approximate Solution [5])

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 $P = NP$).

Outline

- 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

Revision: Minimum submodular set cover (MinCover)

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.

Theorem ([7])

Given a submodular integer-valued function f and a fixed $c \in \mathbf{Z}$, $c \leq f(V)$. Let S_l be the greedy solution and let ℓ be the smallest integer such that $f(S_l) \geq c$. Then

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

Simple trick

MinCover: For any given c solve:

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

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

How can we transform our *multi-objective problem* into the problem with a single objective?

Definition

$$\hat{f}_{i,c}(S) := \min\{f_i(S), c\} \qquad \bar{f}_c(S) := \frac{1}{m} \sum_i \hat{f}_{i,c}(S) \tag{7}$$

Now, we can rewrite the *MinCover* problem as:

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

Simple trick

MinCover: For any given c solve:

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

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

How can we transform our *multi-objective problem* into the problem with a single objective?

Definition

$$\hat{f}_{i,c}(S) := \min\{f_i(S), c\} \qquad \bar{f}_c(S) := \frac{1}{m}\sum_i \hat{f}_{i,c}(S) \tag{7}$$

Note: The previous transformations preserve submodularity.

Now, we can rewrite the *MinCover* problem as:

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

GPC (Greedy partial cover) algorithm

MinCover (for a given c):

$$S = \arg\min_{S \subseteq V} |S| \quad \text{subject to} \quad \bar{f}_c(S) = c. \tag{8}$$

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

GPC (Greedy partial cover) algorithm

MinCover (for a given c):

$$S = \arg\min_{S \subseteq V} |S| \quad \text{subject to} \quad \bar{f}_c(S) = c. \tag{8}$$

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

Theorem (Approximation achieved by the GPC algorithm [5])

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 8 , and

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

Outline

- 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

Constraint relaxation

- *RSFMax* problem:

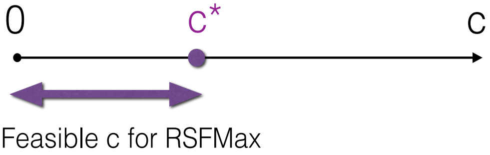
$$\max_{S \in V} \min_i f_i(S), \quad \text{subject to} \quad |S| \leq k$$

Problem (Relaxed RSFMax)

RelRSFMax, the relaxed version of the RSFMax problem:

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

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



Constraint relaxation

- *RSFMax* problem:

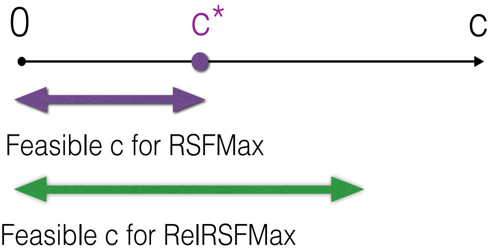
$$\max_{S \in V} \min_i f_i(S), \qquad \text{subject to} \quad |S| \le k$$

Problem (Relaxed RSFMax)

RelRSFMax, the relaxed version of the RSFMax problem:

$$\max_{c,S} c, \qquad \textit{subject to} \quad f_i(S) \ge c \textit{ for } 1 \le i \le m \textit{ and } |S| \le \alpha k$$

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



Algorithm overview

Problem (Relaxed RSFMax)

RelRSFMax, the relaxed version of the RSFMax problem:

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

- *MinCover* problem: for any given c solve:

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

Algorithm overview

Problem (Relaxed RSFMax)

RelRSFMax, the relaxed version of the RSFMax problem:

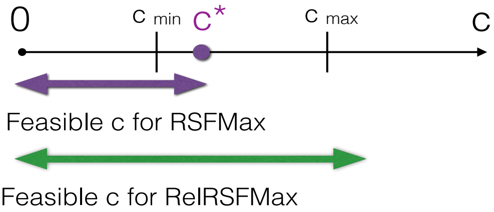
$$\max_{c,S} c, \quad \textit{subject to} \quad f_i(S) \geq c \textit{ for } 1 \leq i \leq m \textit{ and } |S| \leq \alpha k$$

- *MinCover* problem: for any given c solve:

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

- Main idea:
  - 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

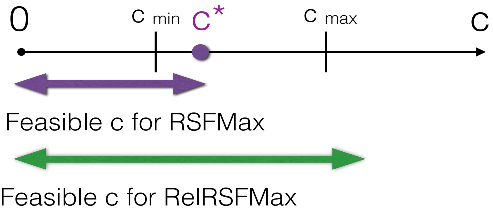
Binary search procedure



Procedure:

- Maintain a lower bound (c_{min}) for *RelRSFMax* and an upper bound for *RSFMax* (c_{max}); Initialize $[c_{\text{min}}, c_{\text{max}}] = [0, \min_i f_i(V)]$
- Successively improve the upper and lower bounds using a binary search procedure
- Invoke the GPC algorithm with $c = (c_{\text{max}} + c_{\text{min}})/2$:

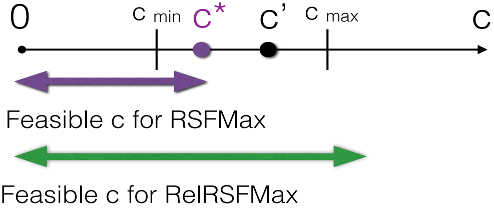
Binary search procedure



Procedure:

- Maintain a lower bound (c_{min}) for *RelRSFMax* and an upper bound for *RSFMax* (c_{max}); Initialize $[c_{\text{min}}, c_{\text{max}}] = [0, \min_i f_i(V)]$
- Successively improve the upper and lower bounds using a binary search procedure
- Invoke the GPC algorithm with $c = (c_{\text{max}} + c_{\text{min}})/2$:
  - $|S_c| > \alpha k$ implies that $c > c^*$, hence c is an upper bound to the RSFMax problem; It is safe to set $c_{\text{max}} = c$
  - $|S_c| \leq \alpha k$ 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 $c_{\text{min}} = c$

Binary search procedure



Procedure:

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

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

Outline

- 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*

The Saturate algorithm

Saturate

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

$c_{\min} \leftarrow 0$; $c_{\max} \leftarrow \min_j f_j(V)$; $S_{\text{best}} \leftarrow \emptyset$

while $(c_{\max} - c_{\min}) > 1/m$ **do**

  $c \leftarrow (c_{\min} + c_{\max})/2$

  $\bar{f}_c(S) \leftarrow \frac{1}{m}\sum_{j=1}^{m} \min\{f_j(S), c\}$

  $\hat{S} \leftarrow \mathrm{GPC}(\bar{f}_c, c)$

  if $|\hat{S}| > \alpha k$ **then**

    $c_{\max} \leftarrow c$

  else

    $c_{\min} \leftarrow c$; $S_{\text{best}} \leftarrow \hat{S}$

return S_{best}

The Saturate algorithm

Saturate

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

```
c_min ← 0; c_max ← min_j f_j(V); S_best ← ∅
while (c_max − c_min) > 1/m do
    c ← (c_min + c_max)/2
    f̄_c(S) ← (1/m) Σ_{j=1}^m min{f_j(S), c}
    Ŝ ← GPC(f̄_c, c)
    if |Ŝ| > αk then
        c_max ← c
    else
        c_min ← c; S_best ← Ŝ
return S_best
```

Theorem (Approximation achieved by the Saturate algorithm [5])

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

The Saturate algorithm

Theorem (Approximation achieved by the Saturate algorithm [5])

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)

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

and

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

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

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

□

Applying *Saturate* to the example problem

$\mathbf{S}$	$\mathbf{f_1(S)}$	$\mathbf{f_2(S)}$	$\min_\mathbf{i} \mathbf{f_i(S)}$
$\emptyset$	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

- 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
- What would *Saturate* choose?

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: $(1, (1-\epsilon)\alpha)$

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

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

References I

[1] B. Bah, S. Becker, V. Cevher, and B. Gözcü.
Metric learning with rank and sparsity constraints.
In *IEEE Int. Conf. Acoustics, Speech, and Sig. Proc. (ICASSP)*, pages 21–25, May 2014.

[2] Luca Baldassarre, Yen-Huan Li, Jonathan Scarlett, Baran Gözcü, Ilija Bogunovic, and Volkan Cevher.
Learning-based compressive subsampling.
http://arxiv.org/abs/1510.06188.

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

[4] Simon Foucart and Holger Rauhut.
A mathematical introduction to compressive sensing.
Springer, 2013.

[5] Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta.
Robust submodular observation selection.
Journal of Machine Learning Research, pages 2761–2801, Dec. 2008.

References II

[6] Ali Sadeghian, Bubacarr Bah, and Volkan Cevher.
Energy-aware adaptive bi-Lipschitz embeddings.
In *Int. Conf. Samp. Theory and Apps.*, Bremen, Germany, 2013.

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