Online Algorithms for Budget-Constrained DR-Submodular Maximization

Omid Sadeghi, Reza Eghbali, Maryam Fazel

UNIVERSITY of WASHINGTON

Online Protocol

- At step $t \in [m]$, $c_t \in \mathbb{R}_+^n$ and $F_t \subset \mathbb{R}_+^n$ arrive online and gradient of $H_i \; \forall i \in [n]$ over the first t variables (i.e., all other $m - t$ variables being zero) is accessible.
- Then, algorithm chooses $x_t \in F_t$.
- **Goal:** Maximize the overall utility while satisfying the linear packing constraints.

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i=1}^{n} H_i(\hat{x}_i) \\ \text{subject to} \quad & x_t \in F_t \quad \forall t \in [m] \\ & \hat{c}_i^T \hat{x}_i \leq 1 \quad \forall i \in [n] \end{aligned}$$

$$X = \begin{bmatrix} | & | & & | & & | \\ x_1 & x_2 & \dots & x_t & \dots & x_m \\ | & | & & | & & | \end{bmatrix}_{n \times m} = \begin{bmatrix} — & \hat{x}_1^T & — \\ — & \hat{x}_2^T & — \\ & \vdots & \\ — & \hat{x}_i^T & — \\ & \vdots & \\ — & \hat{x}_n^T & — \end{bmatrix}_{n \times m}$$

Online Protocol

- At step $t \in [m]$, $c_t \in \mathbb{R}_+^n$ and $F_t \subset \mathbb{R}_+^n$ arrive online and gradient of $H_i \; \forall i \in [n]$ over the first t variables (i.e., all other $m - t$ variables being zero) is accessible.
- Then, algorithm chooses $x_t \in F_t$.
- **Goal:** Maximize the overall utility while satisfying the linear packing constraints.

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i=1}^{n} H_i(\hat{x}_i) \\ \text{subject to} \quad & x_t \in F_t \quad \forall t \in [m] \\ & \hat{c}_i^T \hat{x}_i \leq 1 \quad \forall i \in [n] \end{aligned}$$

Applications: Online knapsack problem, online generalized assignment problem, online budgeted maximum coverage problem, etc.

Diminishing Returns (DR) Property

Definition (DR-Submodular Functions)

A differentiable function $f : \mathbb{R}_+^m \to \mathbb{R}$ is called DR-submodular if:

$$x \succeq y \Rightarrow \nabla f(x) \preceq \nabla f(y).$$

If f is twice differentiable, equivalent to $[\nabla^2 f]_{ij} \leq 0 \; \forall i, j$.

Diminishing Returns (DR) Property

Definition (DR-Submodular Functions)

A differentiable function $f : \mathbb{R}_+^m \to \mathbb{R}$ is called DR-submodular if:

$$x \succeq y \Rightarrow \nabla f(x) \preceq \nabla f(y).$$

If f is twice differentiable, equivalent to $[\nabla^2 f]_{ij} \leq 0 \; \forall i, j$.

- DR-submodularity is the continuous analogue of submodularity of set functions.
- DR-submodular functions are generally non-concave, but they are concave along non-negative directions.

$$f(x + tv) \leq f(x) + t\langle \nabla f(x), v \rangle, \quad t \in \mathbb{R}, v \in \mathbb{R}_+^m.$$

- We assume that $H_i \; \forall i \in [n]$ is monotone DR-submodular.

Penalty Function Design

Idea: Design problem-tailored penalty functions to enforce the linear packing constraints.

$n = 1$: Generalizes the online knapsack problem.

$$G_1(u) = \begin{cases} -L_1 u & \text{if } 0 \leq u < \frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \\ -\frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \frac{L_1}{e} \left(\frac{U_1 e}{L_1}\right)^u & \text{if } u \geq \frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \end{cases}.$$

$n > 1$: Generalizes the Adwords problem.

$$G_i(u) = \frac{L_i}{(e-1)\ln(1+\frac{U_i(e-1)}{L_i})}\left(1-\left(1+\frac{U_i(e-1)}{L_i}\right)^u\right) + \frac{L_i}{e-1}u,$$

where for all $i \in [n]$, U_i and L_i are defined as follows:

$$U_i = \max_{t\in[m]} \frac{\sup_{x\in\mathbb{R}^m:\ \hat{c}_i^T x=1} \nabla_t H_i(x)}{c_{it}}, L_i = \min_{t\in[m]} \frac{\inf_{x\in\mathbb{R}^m:\ \hat{c}_i^T x\leq 1} \nabla_t H_i(x)}{c_{it}}.$$

Algorithm 1 Generalized Sequential Algorithm

for $t = 1$ **to** m **do**

Initialize $\tilde{x}_t(1) = 0$.

c_t, F_t arrive online.

for $k = 1$ **to** K **do**

$$v_t(k) = \arg\max_{v \in F_t} v^T \begin{bmatrix} \vdots \\ \nabla_t H_i\big(\omega_{it}(k)\big) + c_{it} G_i'\big(\hat{c}_i^T \omega_{it}(k)\big) \\ \vdots \end{bmatrix}.$$

$$\tilde{x}_t(k+1) = \tilde{x}_t(k) + \tfrac{1}{K} v_t(k).$$

end for

Output: $\tilde{x}_t = \tilde{x}_t(K+1)$.

end for

where $\omega_{it}(k) := \big[[\tilde{x}_1]_i, \ldots, [\tilde{x}_{t-1}]_i, [\tilde{x}_t(k)]_i, 0, \ldots, 0\big]^T$.

Competitive Ratio Bounds

$n > 1$: CR $\geq \frac{1}{-\min_{i\in[n]} \alpha_{H_i} + \frac{e}{e-1} \max_{i\in[n]} \gamma_i}$, $\gamma_i := \ln(1 + \frac{U_i(e-1)}{L_i})$

$n = 1$: CR $\geq \frac{1}{1-\alpha_{H_1} + \ln(\frac{U_1}{L_1})}$

where $\alpha_{H_i} := \inf_{u:\hat{c}_i^T u \leq 1} \frac{\langle \nabla H_i(u), u \rangle}{H_i(u)} - 1$ captures the curvature of H_i.

- Adwords problem ($\alpha_{H_i} = 0, L_i = U_i = 1 \; \forall i \in [n]$): CR $\geq 1 - \frac{1}{e}$.
- Online linear programming ($\alpha_{H_i} = 0 \; \forall i \in [n]$):
  CR $\geq (1 - \frac{1}{e}) / \big(\max_{i\in[n]} \ln(1 + \frac{U_i(e-1)}{L_i}) \big)$
- Online linear knapsack problem ($n = 1, \alpha_{H_1} = 0$): CR $\geq \frac{1}{1+\ln(\frac{U_1}{L_1})}$