1 Outline

In this lecture, we cover

- submodular function maximization.

2 Submodular Function Maximization

In this section, we consider the problem of maximizing a submodular function. Let $f : 2^{|E|} \to \mathbb{R}_+$ be a **nonnegative** submodular set function. We focus on the case where f is **monotone**, meaning that $f(S) \leq f(T)$ if $S \subseteq T$. Then the problem is given by

$$\text{maximize} \quad f(S) \quad \text{subject to} \quad S \in \mathcal{F} \tag{18.1}$$

where $\mathcal{F} \subseteq 2^{|E|}$ is the collection of feasible subsets. What follows is a list of examples for $\mathcal{F}$.

- Cardinality constraint: $\mathcal{F} = \{S \subseteq E : \ |S| \leq k\}$.
- Knapsack constraint: $\mathcal{F} = \left\{S \subseteq E : \ \sum_{i \in S} c_i \leq B\right\}$.
- Matroid: $\mathcal{F} = \{S \subseteq E : \ S \text{ is an independent set of matroid } \mathcal{M}\}$.

Here, a matroid $\mathcal{M} = (E, \mathcal{I})$ is a pair of a finite ground set E and a collection of independent sets $\mathcal{I} \subseteq 2^E$ satisfying the following properties:

- $\emptyset \in \mathcal{I}$.
- If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$ (hereditary property).
- If $A, B \in \mathcal{I}$ and $|A| < |B|$, then there exists an element $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$ (augmentation property).

For example, the uniform matroid is defined as $\mathcal{I} = \{S \subseteq E : \ |S| \leq k\}$ for some integer k. An extension is the partition matroid where the ground set E is partitioned into disjoint subsets $E_1, E_2, \ldots, E_m$ and the independent sets are defined as

$$\mathcal{I} = \{S \subseteq E : \ |S \cap E_i| \leq b_i \text{ for } i = 1, 2, \ldots, m\}$$

for some given integers $b_1, b_2, \ldots, b_m$. Another example is the graphic matroid, which is defined on a graph $G = (V, E)$ where the ground set is the edge set E and the independent sets are the acyclic subsets of E (i.e., forests).

Example 18.1 (Submodular Welfare Maximization)**.** Suppose that there are m players and a set E of items. Each player $i \in \{1, 2, \ldots, m\}$ has a nonnegative monotone submodular utility function $f_i : 2^E \to \mathbb{R}_+$. The goal is to allocate the items to the players to maximize the total utility:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} f_i(S_i) \\ \text{subject to} \quad & S_i \cap S_j = \emptyset \quad \forall i \neq j, \\ & \bigcup_{i=1}^{m} S_i = E. \end{aligned}$$

This problem can be formulated as a submodular function maximization problem under a partition matroid constraint. To see this, define a new ground set $E' = E \times \{1, 2, \ldots, m\}$. Then E' can be partitioned into $|E|$ subsets, given by

$$E'_e = \{e\} \times \{1, 2, \ldots, m\}$$

for $e \in E$. Next, we define a new submodular function $f : 2^{E'} \to \mathbb{R}_+$ as

$$f(S) = \sum_{i=1}^{m} f_i \left(\{e \in E : \ (e, i) \in S\}\right)$$

for $S \subseteq E'$. Then, the feasible sets can be expressed as

$$\mathcal{F} = \left\{ S \subseteq E' : \ \left| S \cap \left(\underbrace{\{e\} \times \{1, 2, \ldots, m\}}_{E'_e} \right) \right| \leq 1 \text{ for all } e \in E \right\},$$

which corresponds to a partition matroid. Thus, the submodular welfare maximization problem can be written as

$$\begin{aligned} \text{maximize} \quad & f(S) \\ \text{subject to} \quad & S \in \mathcal{F}. \end{aligned} \tag{18.2}$$

□

There are a wide range of applications in business operations, such as online freelancing platforms [SVY20], team selection - sports teams, online gaming [KR15], assortment selection in online shopping websites [Udw21], and influence maximization [KKT03]. Recently, submodular function maximization is applied to many machine learning problems, including feature and variable selection [KG05], dictionary learning [DK11], document summarization [LB10, LB11], image summarization [TIWB14, MJK18], and active set selection in non-parametric learning [MKSK16].

Submodular function maximization (SFM) is NP-hard even with a monotone objective subject to a cardinality constraint [CFN77]. However, there exist polynomial time constant approximation algorithms. We say that a solution $\bar{S} \in \mathcal{F}$ is α-approximate for some $\alpha \in [0, 1]$ if

$$f(\bar{S}) \geq \alpha \cdot \max_{S \in \mathcal{F}} f(S).$$

An α-approximation algorithm would always find an α'-approximate solution for some $\alpha' \geq \alpha$ for every instance of SFM. In other words, the worst-case guarantee is always as good as α times the optimal value. A constant approximation algorithm is an α-approximation algorithm for some fixed $\alpha > 0$.

Let us explain a simple greedy algorithm by Nemhauser, Wolsey, and Fisher [NWF78] that guarantees an $(1-1/e)$-approximate solution. The idea is that until we reach the size limit, we take an element that achieves the maximum marginal return value.

Algorithm 1 Greedy algorithm for submodular maximization

Initialize $S = \emptyset$
while $|S| < k$ **do**
  Take an element $e \in \arg\max \{f(S \cup \{e\}) - f(S) : \; e \in E \setminus S\}$
end while
Return S

Theorem 18.2 (Nemhauser, Wolsey, and Fisher [NWF78]). *Let $f : 2^E \to \mathbb{R}_+$ be a nonnegative monotone submodular function. Let $\bar{S} \subseteq E$ be the outcome of Algorithm 1. Then*

$$f(\bar{S}) \geq \left(1 - \frac{1}{e}\right) \max \{f(S) : |S| \leq k\}.$$

Proof. Let S^* denote an optimal solution to (18.1). Suppose that $e_1, \ldots, e_k$ is the sequence of elements selected by the algorithm. For $i \in \{1, \ldots, k\}$, we use notations $S_0 = \emptyset$ and $S_i = \{e_1, \ldots, e_i\}$ for $i = 1, \ldots, k$. Then $\bar{S} = S_k$.

Let us prove by induction that

$$f(S^*) - f(S_i) \leq \left(1 - \frac{1}{k}\right)^i f(S^*)$$

for all $i \in \{0, 1, \ldots, k\}$. The claim holds when $i = 0$ because f is nonnegative and thus $f(\emptyset) \geq 0$. Suppose that $f(S^*) - f(S_{i-1}) \leq (1 - 1/k)^{i-1} f(S^*)$ for some $i \in \{1, \ldots, k\}$. Since f is submodular, f satisfies the diminishing marginal returns property, so we have

$$f(S^*) - f(S_{i-1}) \leq \sum_{e \in S^* \setminus S_{i-1}} \left(f(S_{i-1} \cup \{e\}) - f(S_{i-1})\right).$$

Since e_i has the maximum marginal return after S_{i-1}, it follows that

$$\begin{aligned}
\sum_{e \in S^* \setminus S_{i-1}} \left(f(S_{i-1} \cup \{e\}) - f(S_{i-1})\right) &\leq |S^* \setminus S_{i-1}| \left(f(S_i) - f(S_{i-1})\right) \\
&\leq k \left(f(S_i) - f(S_{i-1})\right).
\end{aligned}$$

Then it follows that

$$\begin{aligned}
f(S^*) - f(S_i) &= f(S^*) - f(S_{i-1}) - \left(f(S_i) - f(S_{i-1})\right) \\
&\leq \left(1 - \frac{1}{k}\right) \left(f(S^*) - f(S_{i-1})\right) \\
&\leq \left(1 - \frac{1}{k}\right)^i f(S^*)
\end{aligned}$$

where the second inequality is due to the induction hypothesis.

As a result, we have

$$f(\bar{S}) = f(S_k) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(S^*) \geq \left(1 - \frac{1}{e}\right) f(S^*),$$

as required. □

In fact, the greedy algorithm can be extended to an $(1 - 1/e)$-approximation algorithm for the knapsack constraint setting [Svi04].

3 Matroid constraint

In this section, we explain an $(1 - 1/e)$-approximation algorithm for the matroid constraint setting [CCPV07, Von08]. The outline of the algorithm works as follows.

1. First, obtain a continuous relaxation of the given submodular set function (multilinear relaxation).
2. Next, take a continuous relaxation of the feasible region (polymatroid).
3. Then solve the resulting continuous relaxation of the submodular function maximization problem (continuous greedy algorithm).
4. Lastly, round the fractional solution to obtain a discrete solution (pipage rounding).

Let us explain one by one. Given a submodular set function $f : \{0,1\}^{|E|} \to \mathbb{R}$, we take the **multilinear extension** of f given by

$$F(x) = \sum_{S \subseteq E} f(S) \prod_{e \in S} x_e \prod_{e \notin S} (1 - x_e) \quad \text{for } x \in [0,1]^{|E|}.$$

Note that we have

$$F(\mathbf{1}_S) = f(S) \text{ for every } S \subseteq E.$$

Moreover,

$$F(x) = \mathbb{E}_{S \sim x}\left[f(S)\right]$$

where $S \sim x$ means sampling S by selecting each e with probility x_e. The matroid constraint set is given by

$$\mathcal{F} = \{S \subseteq V : \ S \text{ is an independent set of matroid } \mathcal{M}\}.$$

What is the right continuous relaxation for $\mathcal{F}$? For $S \subseteq E$, $\mathrm{rank}(S)$ is defined by the max size of an independent set in S. Then the **polymatroid** of matroid $\mathcal{M}$ is given by

$$P = \left\{ x \in [0,1]^{|E|} : \ \sum_{e \in S} x_e \leq \mathrm{rank}(S) \quad \forall S \subseteq E \right\}.$$

Here, $P \cap \{0,1\}^{|E|} = \mathcal{F}$. Given the multilinear extension of f and the polymatroid P, we obtain the continuous relaxation given by

$$\text{maximize} \quad F(x) \quad \text{subject to} \quad x \in P.$$

Let us present an algorithm that solves the continuous relaxation.

Algorithm 2 Continuous greedy algorithm [CCPV07, Von08]

Input: multilinear extension F and polymatroid P
Start with $x(0) = 0$.
Let $v(x) = \arg\max_{v \in P} \left\{ \nabla F(x)^\top v \right\}$.
Set $dx/dt = v(x)$.
Output $x(1)$.

Theorem 18.3 ([CCPV07, Von08]). *Let $x(t)$ for $t \in [0,1]$ denote the trajectory of the continuous greedy algorithm (2). Then $x(1) \in P$ and*

$$F(x(1)) \geq \left(1 - \frac{1}{e}\right) \max_{S \in \mathcal{F}} f(S).$$

By the theorem, the objective value by Algorithm 2 for the continuous relaxation is at least an $(1 - 1/e)$-approximation of the maximum value of the submodular set function subject to the matroid constraint.

The last step is to round the solution $x(1)$ which potentially has fractional components.

Theorem 18.4 ([AS04, CCPV07]). *There is a randomized polynomial time algorithm that given any $x \in P$, returns $S \in \mathcal{F}$ with*

$$\mathbb{E}\left[f(S)\right] \geq F(x).$$

Together with the previous theorem,

$$\mathbb{E}\left[f(S)\right] \geq F(x(1)) \geq \left(1 - \frac{1}{e}\right) \max_{S \in \mathcal{F}} f(S).$$

References

[AS04] A.A. Ageev and M.I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8:307–328, 2004. 18.4

[CCPV07] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, pages 182–196, 2007. 3, 2, 18.3, 18.4

[CFN77] G. Cornuéjols, M.L. Fisher, and G.L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithm. *Management Science*, 23:789–810, 1977. 2

[DK11] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *International Conference on International Conference on Machine Learning (ICML)*, pages 1057–1064, 2011. 2

[KG05] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 324–331, 2005. 2

[KKT03] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 137–146, 2003. 2

[KR15] Jon Kleinberg and Maithra Raghu. Team performance with test scores. In *ACM Conference on Economics and Computation (EC)*, pages 511–528, 2015. 2

[LB10] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pages 912–920, 2010. 2

[LB11] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pages 510–520, 2011. 2

[MJK18] Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1379–1386, 2018. 2

[MKSK16] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *Journal of Machine Learning Research*, 17(1):1–44, 2016. 2

[NWF78] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14:265–294, 1978. 2, 18.2

[Svi04] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43, 2004. 2

[SVY20] Shreyas Sekar, Milan Vojnovic, and Se-Young Yun. A test score based approach to stochastic submodular optimization. *Management Science*, 2020. 2

[TIWB14] Sebastian Tschiatschek, Rishabh Iyer, Haochen Wei, and Jeff Bilmes. Learning mixtures of submodular functions for image collection summarization. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2014. 2

[Udw21] Rajan Udwani. Submodular order functions and assortment optimization. Technical report, University of California, Berkeley, 2021. 2

[Von08] Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *ACM Symposium on Theory of Computing (STOC)*, page 67–74, 2008. 3, 2, 18.3