ADFOCS 2024, MPI Summer School
Exercise Set: Algorithmic Contract Design

Michal Feldman

August 2024

1 Preliminaries

Set Functions and Oracle Access. Given a set A of n elements, a set function $f : 2^A \to \mathbb{R}^+$ assigns some real *value* to every subset of A, where $f(X)$ denotes the value of $X \subseteq A$. Assume that f is monotone. The *marginal* value of a set X given a set Y is denoted by $f(X \mid Y)$, and defined as $f(X \mid Y) = f(X \cup Y) - f(Y)$. When X is a singleton, we sometimes abuse notation and omit the brackets, i.e., for the marginal value of $X = \{j\}$ given Y, we write $f(j \mid Y)$.

Definition 1.1. *Let A be a set of size n. A set function $f : 2^A \to \mathbb{R}^+$ is said to be:*

- Additive *if there exist $f_1, \ldots, f_n \in \mathbb{R}^+$ such that $f(S) = \sum_{i \in S} f_i$ for every set $S \subseteq A$.*
- Gross substitutes (GS) *if it is submodular (see below) and it satisfies the following triplet condition: for any set $S \subseteq A$, and any three elements $i, j, k \notin S$, it holds that*

$$f(i \mid S) + f(\{j,k\} \mid S) \leq \max\left(f(j \mid S) + f(\{i,k\} \mid S), f(k \mid S) + f(\{i,j\} \mid S)\right).$$

- Budget additive *(BA) if there exist $f_1, \ldots, f_n \in \mathbb{R}^+$ and a budget $B \in [0,1]$ such that for every $S \subseteq A$, $f(S) = \min\{B, \sum_{i \in S} f_i\}$.*
- Submodular *if for any two sets $S \subseteq T \subseteq A$, and any element $j \notin T$, $f(j \mid T) \leq f(j \mid S)$.*
- XOS *if it is a maximum over additive functions. That is, there exists a set of additive functions $f_1, \ldots, f_\ell$ such that for every set $S \subseteq A$, $f(S) = \max_{i \in [\ell]} (f_i(S))$.*
- Subadditive *if for any two sets $S, T \subseteq A$, it holds that $f(S) + f(T) \geq f(S \cup T)$.*
- Supermodular *if for any two sets $S \subseteq T \subseteq [n]$, and any action $j \notin T$, $f(j \mid T) \geq f(j \mid S)$*

All classes above are complement-free except for the supermodular class. It is well known that *Additive* $\subset$ *GS* $\subset$ *Submodular* $\subset$ *XOS* $\subset$ *Subadditive*, with strict containment relations. In addition, *BA* $\subset$ *Submodular*.

Since f is typically of exponential size, it is standard to consider two primitives by which we can access f, defined by the following types of queries:

- A *value* query receives a set $S \subset A$ and returns $f(S)$.
- A *demand* query receives a vector of prices $p = (p_1, \ldots, p_n) \in \mathbb{R}^n_{\geq 0}$, and returns a set S that maximizes $f(S) - \sum_{i \in S} p_i$.

2 Exercises: Combinatorial contracts

Exercise 2.1. *Let $S_\alpha, S_\beta \subseteq A$ be two different sets that maximize the agent's utility for two different contracts $0 \leq \alpha < \beta \leq 1$. Then,*

1. *$f(S_\alpha) < f(S_\beta)$*
2. *$c(S_\alpha) < c(S_\beta)$*

Exercise 2.2. *Consider a single-agent combinatorial actions setting. Prove that any setting with an additive f admits at most n critical points. Find the critical points.*

Exercise 2.3. *Consider a single-agent combinatorial actions setting. Prove that any setting with a supermodular f admits at most n critical points. (Hint: show that for any two contracts $\alpha < \alpha'$ and two corresponding sets in the agent's demand S_α, $S_{\alpha'}$ it holds that $S_\alpha \subseteq S_{\alpha'}$.)*

Exercise 2.4. *Consider a single-agent combinatorial actions setting. Prove that the optimal contract problem for budget additive success probability is NP-hard.*

Hint: construct a reduction from SUBSET-SUM. *Subset-sum receives as input a (multi-)set of positive integer values $X = \{x_1, \ldots, x_n\}$ and an integer value Z. The question is whether there exists a subset $S \subseteq X$ such that $\sum_{j \in S} x_j = Z$. W.l.o.g., assume that $x_i < Z$ for all i (all numbers greater than Z can be ignored), and that $\sum_{i \in X} x_i > Z$ (otherwise this is an easy instance).*

Exercise 2.5. *Prove the correctness of the recursive algorithm for enumerating all critical points in poly-time, given access to a demand oracle.*

3 Exercises: Ambiguous contracts

Exercise 3.1. *Prove that in the following example, the ambiguity gap is unbounded.*

rewards:	$r_1 = -r$	$r_2 = -r$	$r_3 = 0$	$r_4 = r$	*costs*
action 1:	0	0	1	0	$c_1 = 0$
action 2:	0.5	0	0	0.5	$c_2 = 10$
action 3:	0	0.5	0	0.5	$c_3 = 10$
action 4:	0.2	0.2	0	0.6	$c_4 = 20$

Exercise 3.2. *Prove that in the following example the principal gains from using an ambiguous contract by implementing action 6, which cannot be implemented with a classic contract.*

rewards:	$r_1 = -200$	$r_2 = 0$	$r_3 = 21$	$r_4 = 21$	*costs*
action 1:	0	1	0	0	$c_1 = 0$
action 2:	0.1	0	0.9	0	$c_2 = 8$
action 3:	0.1	0	0	0.9	$c_3 = 8$
action 4:	0	0	1	0	$c_4 = 10$
action 5:	0	0	0	1	$c_5 = 10$
action 6:	0	0	0.5	0.5	$c_6 = 11$

Exercise 3.3. *Prove that the algorithm shown in class for computing the optimal ambiguous contract implementing action i indeed implements action i.*