ADFOCS 2024, MPI Summer School
Exercise Set (w/ Solutions): 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 $f_1, \dots, f_n \in \mathbb{R}^+$ 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 $f_1, \dots, f_n \in \mathbb{R}^+$ 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 $f_1, \dots, f_\ell$ 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 $p = (p_1, \dots, p_n) \in \mathbb{R}^n_{\geq 0}$, 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. $c(S_\alpha) < c(S_\beta)$

Solution: The fact that S_α is superior for contract α and S_β is superior for β implies that

$$\alpha f(S_\beta) - c(S_\beta) \leq \alpha f(S_\alpha) - c(S_\alpha)$$
$$\beta f(S_\beta) - c(S_\beta) \geq \beta f(S_\alpha) - c(S_\alpha)$$

Rearranging,

$$\alpha[f(S_\beta) - f(S_\alpha)] \leq \beta[f(S_\beta) - f(S_\alpha)]$$

Together with the fact that $\alpha < \beta$ we get that $f(S_\alpha) \leq f(S_\beta)$. Due to consistent tie breaking, it must be that $f(S_\alpha) < f(S_\beta)$. For the second part of the claim, observe that if $c(S_\alpha) \geq c(S_\beta)$, then

$$u(\alpha, S_\beta) = \alpha f(S_\beta) - c(S_\beta) \geq \alpha f(S_\alpha) - c(S_\alpha) = u(\alpha, S_\alpha)$$

Which contradicts the optimality of S_α.

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.*

Solution: for every scenario with an additive f, every action i belongs to the agent's best response if and only if $\alpha \geq c_i/f(i)$, independent of the other actions. This is the point α satisfying $\alpha f(S \cup i) - c_i = \alpha f(S)$, independent of the set S. Thus, there are at most n *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 $S_\alpha \subseteq S_{\alpha'}$.)*

Solution: We show that for any two contracts $\alpha < \alpha'$ and two corresponding sets in the agent's demand S_α, $S_{\alpha'}$ it holds that $S_\alpha \subseteq S_{\alpha'}$. This holds for any setting with supermodular f and submodular c. The proof follows.

If $S_\alpha = S_{\alpha'}$ the claim obviously hold. Otherwise, assume that $S_{\alpha'}$ is a maximal best-response for contract α' (this is in line with our tie-breaking assumption), and also that $S_\alpha \setminus S_{\alpha'} = R$ is such that $R \neq \emptyset$, we will show that a contradiction is reached. By the fact that S_α is optimal for α, it must be that

$$u_a(\alpha, R \mid S_\alpha \cap S_{\alpha'}) = u_a(\alpha, S_\alpha) - u_a(\alpha, S_\alpha \cap S_{\alpha'}) \geq 0$$

By the supermodularity of f and submodularity of c it holds that $f(R \mid S_\alpha \cap S_{\alpha'}) \leq f(R \mid S_{\alpha'})$ and $c(R \mid S_\alpha \cap S_{\alpha'}) \geq c(R \mid S_{\alpha'})$. Putting everything together we get

$$\begin{aligned}
u_a(\alpha', R \mid S_{\alpha'}) &= \alpha' f(R \mid S_{\alpha'}) - c(R \mid S_{\alpha'}) \\
&\geq \alpha' f(R \mid S_\alpha \cap S_{\alpha'}) - c(R \mid S_\alpha \cap S_{\alpha'}) \\
&\geq \alpha f(R \mid S_\alpha \cap S_{\alpha'}) - c(R \mid S_\alpha \cap S_{\alpha'}) \\
&= u_a(\alpha, R \mid S_\alpha) \\
&\geq 0,
\end{aligned}$$

where the second inequality follows from the monotonicity of f, which imply $f(R \mid S_\alpha \cap S_{\alpha'}) \geq 0$. Thus, we can add R to $S_{\alpha'}$ while not losing utility, contradicting its maximality.

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).*

Solution: We prove the hardness by 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).

Given an instance $(x_1, \ldots, x_n, Z)$ to subset-sum, construct an instance to the optimal contract problem for budget additive functions over n actions as follows. For every action $i = 1, \ldots, n$, set $f(\{i\}) = x_i$, and set $B = Z$. I.e., for every set S, $f(S) = \min(Z, \sum_{i \in S} x_i)$. Let the cost function be $c(i) = \epsilon \cdot x_i$, where $\epsilon = \frac{1}{Z^2}$.

If there exists a set S such that $\sum_{i \in S} x_i = Z$, then for a contract of $\alpha \geq \epsilon$ the agent's best-response is the set S, and for $\alpha < \epsilon$ the agent's best response is the empty-set. Thus, the optimal contract is to set $\alpha = \epsilon$ where the principal utility is $(1 - \epsilon) \cdot Z$.

Consider next the case where there does not exist a set S such that $\sum_{i \in S} x_i = Z$. Let $Z_1 = argmin\{z > Z \mid \exists S \subseteq [n]. \sum_{i \in S} x_i = z\}$, and let S_1 be the set that sums to Z_1. Similarly, let $Z_2 = \arg\max\{z < Z \mid \exists S \subseteq [n]. \sum_{i \in S} x_i = z\}$, and let S_2 be the set that sums to Z_2.

Every set S such that $\sum_{i \in S} x_i > Z$ gives an agent's utility of $\alpha Z - \epsilon \sum_{i \in S} x_i$. Thus, S_1 is optimal among all these sets. Similarly, every set S such that $\sum_{i \in S} x_i < Z$ gives an agent's utility of $(\alpha - \epsilon) \sum_{i \in S} x_i$. Thus, for $\alpha \geq \epsilon$, S_2 is optimal among all these sets. It follows that there are exactly two critical α's, namely $\alpha_1 = \epsilon$, where the agent selects S_2 and the principal's utility is $(1 - \epsilon) Z_2$, and $\alpha_2 = \frac{Z_1 - Z_2}{Z - Z_2} \cdot \epsilon$, where the agent selects S_1 and the principal's utility is $(1 - \frac{Z_1 - Z_2}{Z - Z_2} \cdot \epsilon) Z$.

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

Solution: We prove that for any segment $[\alpha, \beta]$, $\texttt{CV}(\alpha, \beta)$ returns all the critical values in that segment.

We prove it by induction over the number of critical values in the segment.

For the basis of the induction, assume there are no critical values in $[\alpha, \beta]$. Thus, it must be that the agent's utility is linear in that segment, by the consistency of tie breaking $S_\alpha = S_\beta$ — and the algorithm returns $\emptyset$.

If there exists a single critical value, since the agent's utility is piece-wise linear, $S_\alpha \neq S_\beta$ and the critical value is at the intersection of the linear functions $u_a(x, S_\alpha) = x f(S_\alpha) - c(S_\alpha)$ and $u_a(x, S_\beta) = x f(S_\beta) - c(S_\beta)$, which is exactly $\gamma = \frac{c(S_\beta) - c(S_\alpha)}{f(S_\beta) - f(S_\alpha)}$. Note that this implies that $S_\gamma = S_\beta$, as ties are broken in favor of the principal.

Assume the correctness of the algorithm for any a segment with $k > i \geq 1$ critical values and consider the segment $[\alpha, \beta]$ with k critical values. The inductive step will show that there is at least one critical value on each side of γ, so each of the sub-segments $[\alpha, \gamma]$ and $[\gamma, \beta]$ have strictly less than k critical values. Thus, when calling $\texttt{CV}(\alpha, \gamma)$ and $\texttt{CV}(\gamma, \beta)$, by the induction hypothesis, the critical values for each of these sub-segments are returned. As implied by the base of the induction, if γ is a critical value it will emerge from the $\texttt{CV}(\alpha, \gamma)$ branch in the calls tree.

Let $\delta_{\min}$ be the smallest critical value in $(\alpha, \beta]$. As $S_{\delta_{\min}}$ is the indifference point between S_α and $S_{\delta_{\min}}$, $\delta_{\min} = \frac{c(S_{\delta_{\min}}) - c(S_\alpha)}{f(S_{\delta_{\min}}) - f(S_\alpha)}$. Aiming for contradiction, assume $\delta_{\min} \in [\gamma, \beta]$. Thus, S_α dominates $S_{\delta_{\min}}$ in the segment $[\alpha, \gamma)$ and at γ:

$$u_a(S_\beta, \gamma) = u_a(S_\alpha, \gamma) \geq u_a(S_{\delta_{\min}}, \gamma)$$

By the monotonicity lemma $f(S_\beta) > f(S_{\delta_{\min}})$ and so S_β dominates $S_{\delta_{\min}}$ in $[\gamma, \beta]$, contradicting the optimality of $S_{\delta_{\min}}$.

An analogous argument can be applied to show the existence of a critical value in the segment $(\gamma, \beta]$: let $\delta_{\max}$ be the largest critical value in $[\alpha, \beta]$. Observe that $S_{\delta_{\max}} = S_\beta$, and let S be the optimal set just preceding S_β (that is, for every sufficiently small $\epsilon > 0$, $S_{\delta_{\max}-\epsilon} = S$). Aiming for contradiction, assume $\delta_{\max} \in [\alpha, \gamma]$. As $\delta_{\max}$ is the indifference point between S and S_β, and $f(S_\beta) > f(S)$, it holds that S_β dominates S in the segment $[\gamma, \beta]$. Under contract γ:

$$u_a(S_\alpha, \gamma) = u_a(S_\beta, \gamma) \geq u_a(S, \gamma)$$

Combined with $f(S) > f(S_\alpha)$ (by Lemma **??**), we get that for any $\delta < \gamma$:

$$\begin{aligned} u_a(S_\alpha, \delta) &= u_a(S_\alpha, \gamma) - (\gamma - \delta) f(S_\alpha) \\ &\geq u_a(S, \gamma) - (\gamma - \delta) f(S_\alpha) \\ &> u_a(S, \gamma) - (\gamma - \delta) f(S) \\ &= u_a(S, \delta) \end{aligned}$$

This implies that S in never optimal, a contradiction.

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	$c_4 = 20$

Solution: Actions 2 and 3 generate negative welfare, and hence only action 4 is capable (depending on r) of producing positive welfare. Welfare is given by

$$\max\{0, 0.2r - 20\},$$

and is positive if and only if $r > 100$. An optimal classic contract implementing action 4 is $\langle t, 4\rangle = \langle (0, 0, 0, 100), 4\rangle$, which gives

$$U_P(\langle t, 4\rangle) = 0.2r - 60,$$

which is positive if and only if $r > 300$. An optimal ambiguous contract implementing action 4 is $\langle \tau, 4\rangle = \langle \{t^1, t^2\}, 4\rangle = \langle \{(100, 0, 0, 0), (0, 100, 0, 0)\}, 4\rangle$, giving

$$U_P(\langle \tau, 4\rangle) = 0.2r - 20,$$

which is positive if and only if $r > 100$. Hence for $r \in (100, 300]$, the best classic contract generates a payoff of 0, while the best ambiguous contract generates a positive payoff, yielding an infinite ambiguity gap.

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	$c_6 = 11$

Solution: Action 6 cannot be implemented by a classic contract, with the half/half combination of actions 4 and 5 giving the same distribution over outcomes at a lower cost. Actions 2 and 3 have a negative expected welfare, and so will never be optimal for the principal. Actions 4 and 5 can both be implemented with a classic contract, and yield the same maximal utility for the principal. Optimal classic contracts for these actions include $\langle(0, 0, 20, 0), 4\rangle$ and $\langle(0, 0, 0, 20), 5\rangle$, each giving the principal an expected utility of 1. In contrast, the optimal ambiguous contract $\langle\{(0, 0, 22, 0),(0, 0, 0, 22)\}, 6\rangle$ implements action 6, for an expected utility of 10.

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

Solution: We first show that if there exists an action $i' \neq i$ such that $p_{i'} = p_i$ and $c_{i'} < c_i$, then it is impossible to implement action i with an ambiguous contract. For the sake of contradiction, suppose that ambiguous contract $\langle\tau, i\rangle = \langle\{t^1, \dots, t^k\}, i\rangle$ implements action i. In this case, since $p_i = p_{i'}$, we have $T_i(t^\ell) = T_{i'}(t^{\ell'})$ for all $\ell, \ell' \in [k]$. But then $U_A(i' \mid \tau) = \min_{\ell\in[k]} T_{i'}(t^\ell)-c_{i'} > \min_{\ell\in[k]} T_i(t^\ell)-c_i = U_A(i \mid \tau)$, contradicting the fact that $\langle\tau, i\rangle$ is incentive compatible.

Next we show that if there is no action $i' \neq i$ such that $p_{i'} = p_i$ and $c_{i'} < c_i$, then action i can be implemented with an ambiguous contract. In this case, for each action $i' \neq i$, either (i) $p_{i'} \neq p_i$ or (ii) $p_{i'} = p_i$ and $c_{i'} \geq c_i$. Let A be the actions of type (i). If A is empty, then i must be a zero-cost action. A (consistent) ambiguous contract for implementing that action is $\langle\tau, i\rangle$ with $\tau = \{(0, \dots, 0)\}$.

Assume A is nonempty. We construct an ambiguous contract $\langle\tau, i\rangle$ for implementing action i that has one contract $t^{i'}$ for each action $i' \neq i$ of type (i). For each action $i' \in A$, let $j(i')$ be an outcome j such that $p_{ij}/p_{i'j}$ is maximal. Note that $p_{ij(i')}/p_{i'j(i')} > 1$. Let

$$T = \max_{i'\in A}\left\{\min\left\{x \geq 0 \;\middle|\; p_{ij(i')} \cdot \frac{x}{p_{ij(i')}} - c_i \geq p_{i'j(i')} \cdot \frac{x}{p_{ij(i')}} - c_{i'}\right\}\right\}.$$

For each $i' \in A$, let $t^{i'}_{j(i')} = T/p_{ij(i')}$ and $t^{i'}_{j'} = 0$ for $j' \neq j(i')$.

We conclude by verifying that $\langle\tau, i\rangle = \langle\{t^{i'} \mid i' \in A\}, i\rangle$ is a (consistent) ambiguous contract that implements action i. It is easy to check consistency.

To see that $\langle\tau, i\rangle$ is incentive compatible, first consider actions $i' \neq i$ of type (ii). For these actions we have

$$U_A(i' \mid \tau) = U_A(i \mid \tau) + c_i - c_{i'} \leq U_A(i \mid \tau),$$

where we used that $p_{i'} = p_i$ and $c_{i'} \geq c_i$.

Next consider actions $i' \neq i$ of type (i). For these actions, there must be a $T_{i'} \geq 0$ with $T_{i'} \leq T$ such that

$$p_{ij(i')} \cdot \frac{T_{i'}}{p_{ij(i')}} - c_i \geq p_{i'j(i')} \cdot \frac{T_{i'}}{p_{ij(i')}} - c_{i'}.$$

Since $T_{i'} \leq T$ and $p_{ij(i')} > p_{i'j(i')}$ this implies

$$\begin{aligned} U_A(i \mid \tau) = p_{ij(i')} \cdot \frac{T}{p_{ij(i')}} - c_i &\geq p_{i'j(i')} \cdot \frac{T}{p_{ij(i')}} - c_{i'} \\ &= \min_{i''\in A}\left(p_{i'j(i'')} \cdot \frac{T}{p_{ij(i'')}} - c_{i'}\right) = U_A(i' \mid \tau), \end{aligned}$$

where the first equality holds by consistency, the second equality holds by definition of $j(i')$, and the final equality holds by definition.