Notes on Interim Feasibility of Bayesian Mechanisms

Hu Fu

November 16, 2017

1 Border's Inequalities

We are given n bidders whose private types $typei[1], \ldots, t_n$ are drawn independently from distributions $F_1, \cdots, F_n$, respectively, and who are competing for a single item. The set of all Bayesian incentive compatible, interim individually rational mechanisms is easily representable by a polytope if we use $x_i(\boldsymbol{t})$ and $p_i(\boldsymbol{t})$ to denote the allocation and payment of each bidder i. However, if we let T_i be the support of distribution F_i, then the input to this problem is of size $\sum_i |T_i|$, whereas the number of variables is $\prod_i |T_i|$, which can be exponential in n compared with the input length. Alternatively, one could consider using as variables the *interim allocations and payments*: for each bidder i and type t_i, let $x_i(t_i)$ be $\mathbf{E}_{\boldsymbol{t}_{-i}}[x_i(t_i, \boldsymbol{t}_{-i})]$, and $p_i(t_i) := \mathbf{E}_{\boldsymbol{t}_{-i}}[p_i(t_i, \boldsymbol{t}_{-i})]$. Then BIC and interim IR constratins are still fairly easy to express;

$$t_i x_i(t_i) - p_i(t_i) \geq t_i x_i(t_i') - p_i(t_i'), \qquad \forall i, \forall t_i, t_i';$$
$$t_i x_i(t_i) - p_i(t_i) \geq 0, \qquad \forall i, \forall t_i.$$

and so is the expected revenue: $\sum_i \sum_{t_i} p_i(t_i)$. However, the feasibility constraints are trickier. With the ex post allocations, they are easy to express: $\forall i, \forall \boldsymbol{t}, \sum_i x_i(\boldsymbol{t}) \leq 1$. With interim allocations, things are not that obvious.

Theorem 1. *A set of interim allocation rules are feasible if and only if for every $S_i \subseteq T_i$, $i = 1, \cdots, n$,*

$$\sum_i \sum_{t_i \in S_i} x_i(t_i) F_i(t_i) \leq 1 - \prod_i (1 - F_i(S_i)), \tag{1}$$

where $F_i(S_i)$ denotes $\sum_{t \in S_i} F_i(t)$.[1]

The inequalities in Theorem 1 are known as the *Border's inequalities*, as they were first proposed and proved by Border (1991, 2007). The proof given here is equivalent to the approach in Che et al. (2013), but is somewhat simpler.

Proof. The "only if" part is easy. Note that the right hand side of the inequality is the probability with which any type in $\cup_i S_i$ occurs, and the LHS the total allocation to types in $\cup_i S_i$. In order for a mechanism to be feasible, the RHS is obviously an upper bound on the LHS.

For the "if" part, we use a network flow argument. Let's build a graph consisting a source node $S_\perp$, a left group of nodes L, a right group of nodes R, and a sink node $S_\top$. Each node

[1] Throughout this note, we assume w.l.o.g. that the type spaces are disjoint.

in L corresponds to a type $t_i \in T_i$, for some i; each node in R corresponds to a type profile $\boldsymbol{t} = (t_1, \ldots, t_n)$, where $t_i \in T_i, \forall i$. Without causing too much confusion, we will simply refer to the nodes in L and R by their coresponding types and type profiles. There is an edge going from $S_\perp$ to each node in L, with capacity $x_i(t_i)F_i(t_i)$ for the edge going to the type t_i. There is an edge going from each node in R to $S_\top$, with capacity $\prod_i F_i(t_i)$ on the edge leaving the type profile $(t_1, \ldots, t_n)$. For each node $(t_1, \ldots, t_n)$ in R, there are n edges going from the nodes in L that correspond to $t_1, \ldots, t_n$, respectively. The capacity on each of these edges is infinity.

Any implementation of $x_1, \ldots, x_n$ gives rise to a flow in the graph that saturates the first group of edges (going from $S_\perp$ to L). By the max-flow min-cut theorem, this is possible only if these edges constitute a minimum cut of the graph. We will show that this is true precisely when the inequalities in Theorem 1 hold.

Any cut that has $S_\perp$ and $S_\top$ on the same side must contain an edge with infinite capacity (that connects L and R), and cannot be a minimum cut. So assume $S_\perp$ and $S_\top$ are on different sides of the cut. If $S_1 \subseteq T_1, \cdots, S_n \subseteq T_n$ are the nodes in L on the same side with $S_\perp$, in order that the cut does not contain any edge with infinite capacity, the neighbors of $S_1, \cdots, S_n$ in R must be precisely the set of nodes in R that are also on this side. These are the type profiles which contain at least one type from $\cup_i S_i$. Let $\overline{S_i}$ be $T_i \backslash S_i$, then the size of the cut is $\sum_i \sum_{t_i \in \overline{S_i}} x_i(t_i)F_i(t_i) + \mathbf{Pr}[\boldsymbol{t} : \exists i, t_i \in S_i]$. The condition that this cut cannot be the bottleneck of the flow is:

$$\sum_i \sum_{t_i \in T_i} x_i(t_i)F_i(t_i) \leq \sum_i \sum_{t_i \in \overline{S_i}} x_i(t_i)F_i(t_i) + \mathbf{Pr}\left[\boldsymbol{t} : \exists i, t_i \in S_i\right].$$

Rearranging terms, this is just (1). □

A mechanism expressed by its interim allocation and payments rules is said to be in its *reduced form.* Theorem 1 allows us to precisely express the range of feasible mechanisms using only the reduced form.

The number of constraints for interim feasibility, however, is still exponential in the length of the input. With a polynomial number of variables and exponential number of constraints, it is natural to consider using the ellipsoid method for optimization, for which we would need a separation oracle. That is, we need a polynomial-time algorithm that outputs

$$\arg\min_{S_i \subseteq T_i, \forall i}[1 - \prod_i (1 - F_i(S_i))] - \sum_i \sum_{t_i \in S_i} x_i(t_i)F_i(t_i).$$

Furthermore, even after we solve the linear programming with the interim allocations and payments as variables, we would need to be able to implement them. Specifying all the ex post allocations and payments amounts to an output length that is exponential in the input length. What we need is an oracle that, given feasible interim allocation and payment rules, at an input type profile $\boldsymbol{t} = (t_1, \ldots, t_n)$, output ex post allocations and payments for each bidder, so that in expectation the given interim allocations and payments are guaranteed to be implemented. Section 2 provides an algorithm for the separation oracle, and Section 3 discusses the ex post implmentation. In the end, Section 4 gives a few examples (beyond Myerson's setting) to show the power of this approach.

2 Separation Oracle via Submodular Minimization

If we define a function $h : 2^{\cup_i T_i} \to \mathbb{R}$ as $h(\cup_i S_i) = [1 - \prod_i (1 - F_i(S_i))] - \sum_i \sum_{t_i \in S_i} x_i(t_i) F_i(t_i)$, then our task is to minimize this function. We show that h is submodular, and therefore polynomial-time algorithms for submodular minimization would give us a separation oracle.

Definition 1. A set function $f : 2^M \to \mathbb{R}$ is said to be *submodular* if, for all $S, T \subseteq T$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. Further, f is *monotone* if for all $S \subseteq T \subseteq M$ we have $f(S) \leq f(T)$.

Proposition 1. *A function is submodular if and only if, for any $S \subseteq T \subseteq M$ and any $j \in M$, $f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$.*

The property in Proposition 1 is often called *decreasing marginal value.* The following proposition is left as an exercise.

Proposition 2. *The function h is submodular.*

The supplementary reading contains examples of submodular functions and a polynomial-time algorithm for minimizing such functions. Hence we have a fast separation oracle for the Border's inequalities.

3 Ex post Implementation

Describing an ex post implementation given a feasible mechanism in its reduced form may seem like a daunting problem, at least a priori. Absent structures, an ex post implementation generally specifies, for every type profile, an allocation and a payment for each bidder, and this is exponential in the input length, for product distributions. We will see in this section that, due to the submodularity of the h functions, there always exists a succinctly reprensentable ex post implementation, not only for the revenue optimal auction, but for interim feasible mechanism.

Definition 2. Given any submodular function $f : 2^M \to \mathbb{R}$, the *polymatroid* associated with it is $\{\boldsymbol{x} \in \mathbb{R}_+^M | \forall S \subseteq M, \sum_{j \in S} x_j \leq f(S)\}$.

We will write $x(S)$ as a shorthand for $\sum_{j \in S} x_j$. We also assume $f(\emptyset) = 0$ throughout this section.

From (1) in Theorem 1, we see that the set of interim feasible mechanisms is given by the polymatroid associated with the monotone submodular function $1 - \prod_i (1 - F_i(S_i))$. We denote this polymatroid by $\mathcal{P}_B$.

Plan. Given any interim feasible mechanism, which is a point in this polymatroid, we will decompose it into a convex combination of vertices of the polymatroid. Such a decomposition gives rise to a probability distribution over the vertices, which themselves correspond to feasible mechanisms; implementing them according to this probability distribution amounts to implementing the given mechanism. It therefore suffices to show that, (i) such a decomposition can be found efficiently, and (ii) each vertex corresponds to a mechanism whose implementation can be succinctly described.

Note that a fast decomposition implicitly requires that the number of vertices used in the decomposition cannot be too large. The existence of such a decomposition is guaranteed by the well-known Carathéodory theorem:

Theorem 2 (Carathéodory)**.** *If a point $\boldsymbol{x} \in \mathbb{R}^d$ lies in the convex hull of a set T, then $\boldsymbol{x}$ can be written as the convex combination of at most $d+1$ points in T.*

The following is also a well-known result from combinatorial optimization. For completeness, we provide a proof.

Theorem 3. *Each vertex of the polymatroid associated with a monotone submodular function f corresponds to a chain of subsets $\emptyset = S_0 \subseteq S_1 \subseteq S_2 \subseteq \cdots \subseteq S_k \subseteq M$ for some k, such that $|S_r \setminus S_{r-1}| = \{j_r\}$ for $r = 1, \cdots, k$ (j_r is an element in M), and $x_{j_r} = f(S_r) - f(S_{r-1})$ for $r = 1, \cdots, k$, and $x_j = 0$ for any other $j \in M$.*

Proof. We make use of the fact that any vertex of a polytope is an optimal solution to the problem of maximizing some linear objective within the polytope. Consider any linear objective $\sum_{j \in M} a_j x_j$. We show that the following greedy algorithm produces the optimal solution: sort the elements in M by a_j, so that $a_{j_1} \geq a_{j_2} \geq \ldots \geq a_{j_k} \geq 0 > a_{j_{k+1}} \geq \ldots \geq a_{j_{|M|}}$; let S_r be $\{j_1, \ldots, j_r\}$ and $x_i^*[j_r] = f(S_r) - f(S_{r-1})$ for $r = 1, \cdots, k$, and let $x_i^*[j_r] = 0$ for $r > k$.

We first show the feasibility of $\boldsymbol{x}^*$. All its coordinates are nonnegative by the monotonicity of f. We also need to show that for any $S \subseteq M$, $x^*(S) \leq f(S)$. We induct on the size of S. The base case when $S = \emptyset$ is trivially true. Let's assume $x^*(S) \leq f(S)$ for any S of size at most $\ell - 1$. Then for any S of size ℓ, let j_s be the last element in S (last according to the order $j_1, j_2, \ldots, j_|M|$. Then

$$x^*(S) = x^*(S \setminus \{j_s\}) + x_i^*[j_s] \leq f(S \setminus \{j_s\}) + f(S_s) - f(S_{s-1}) \leq f(S),$$

where the first inequality is by the induction hypothesis, and the second by the submodularity of f.

We show the optimality by constructing a dual solution whose value matches that of the solution by the greedy algorithm. Recall that the dual program is

$$\begin{aligned} \min \sum_{S \subseteq M} & f(S) y_S \\ \text{s.t.} \sum_{S \ni j} & y_S \geq a_j, \qquad \forall j \in M; \\ & y_S \geq 0, \qquad \forall S \subseteq M. \end{aligned}$$

Let $y^*_{S_k}$ be a_{j_k}. Then for $r = 1, 2, \cdots, k-1$, let $y^*_{S_r}$ be $a_{j_r} - a_{j_{r+1}}$. For all other $S \subseteq M$, let y^*_S be 0. It is easy to verify that this $\mathbf{y}^*$ is dual feasible. We show optimality of both $\boldsymbol{x}^*$ and $\mathbf{y}^*$ by showing that the dual program's value given by $\mathbf{y}^*$ is equal to primal program's value given by $\boldsymbol{x}^*$:

$$\sum_S f(S) y^*_S = f(S_k) a_{j_k} + \sum_{r=1}^{k-1} f(S_r)(a_{j_r} - a_{j_{r+1}}) = \sum_{r=1}^{k} a_{j_r}(f(S_r) - f(S_{r-1})) = \sum_j a_j x_i^*[j].$$

□

Definition 3. A single-item auction is said to be a *ranking* mechanism if its allocation rule is determined by a total order σ over the types $\cup_i T_i$ and a null type $\perp$: given any type profile, if any type is ranked by σ before $\perp$, the item is allocated to the type ranked first by σ; otherwise, no bidder gets the item.

Proposition 3. *Each vertex of the polymatroid $\mathcal{P}_B$ corresponds to a ranking mechanism.*

Proof. As we saw in the greedy algorithm, any vertex of $\mathcal{P}_B$ corresponds to a chain of subsets of $\cup_i T_i$: $\emptyset = S_0 \subseteq S_1 \subseteq \cdots \subseteq S_k \subseteq \cup_i T_i$. These are the sets of types for which the Border's inequalities (1) are tight. This means that whenever a type from one of these sets appears, that type must be allocated the item. Since $|S_r \setminus S_{r-1}| = 1$ for $r = 1, \cdots, k$, this effectively gives a ranking for the types in S_k: the type in S_1 has precedence over the type in $S_2 \setminus S_1$, which in turn has precedence over the type in $S_3 \setminus S_2$, and so on. The greedy algorithm also sets the interim allocation for all types not in S_k to 0. Therefore these types never receive the item, as specified in the ranking mechanism. □

Combining Theorem 2 and Proposition 3, we get the following characterization of any interim feasible mechanism:

Theorem 4. *Any interim feasible mechanism for selling a single item to n bidders with types drawn independently from type spaces $T_1, \cdots, T_n$ can be implemented by a distribution over $\sum_i |T_i| + 1$ ranking mechanisms.*

4 Applications

We briefly give two natural scenarios of revenue maximization beyond the reach of Myerson's characterization, where we could nonetheless compute and implement the revenue optimal BIC and interim IR auctions.

Example 1. Let there be n bidders bidding for a single item. The bidders' values are drawn independently from known distributions $F_1, \cdots, F_n$, but each bidder i has a publicly known budget B_i. In addition to the interim IR and BIC constraints, the mechanism must never charge bidder i a payment of more than B_i.

Remark 1. In fact, it is not hard to also write an LP for the case of bidders with private budgets, in which case the budget becomes part of the type, drawn from a distribution.

Example 2. Let there be n bidders bidding for m items. Each bidder's type describes an additive valuation, that is, for any subset S of items, bidder i's value is given by $v_i(S) = \sum_{j \in S} v_i(\{j\})$. Each bidder's type is drawn independently from known distributions $F_1, \cdots, F_n$.

In this example, besides the obvious IC and IR constraints, the interim feasibility constraints are simply the concatenation of those for each item.

References

Border, K. C. (1991). Implementation of reduced form auctions: A geometric approach. *Econometrica*, 59(4):1175–87.

Border, K. C. (2007). Reduced form auctions revisited. *Economic Theory*, 31(1):167–181.

Che, Y.-K., Kim, J., and Mierendorff, K. (2013). Generalized reduced-form auctions: A network-flow approach. *Econometrica*, 81(6):2487–2520.