Lecture 25: March 27

Lecturer: Éva Tardos *Scribe: Eyvind Niklasson, Aaron Ferber*

In this lecture we prove that **submodularity implies XOS**. Additionally, we prove that **fractionally subadditive implies XOS**, which shows that **fractionally subadditive is equivalent to XOS**, due to our proof the other way around from last time.

25.1 Previous Work

The past few lectures we've defined and proved containment for various function classes including:

Lemma 25.1 *Subadditivity $\subseteq$ Fractionally Subadditive*

Lemma 25.2 *XOS $\subseteq$ Fractionally Subadditive*

25.2 Proof: Submodular $\subseteq$ XOS (Containment)

We prove this by showing that if a valuation function V is submodular it must be XOS. Recall the definition of XOS

Definition 25.3 (XOS) *v is XOS if we can write it as $v(A) = \max_k \sum_{j \in A} v_j^k$, for valuations on using item i in a specific way k, v_j^k. Note that this is not exponentiating v_j but rather indexing into the valuations with a particular usage k.*

As we want to show the valuation is XOS, we need to define alternate ways to value the items. Assume we have some ordering of the items. Let's call this π. Now, just for fun, let's define:

$S^{\pi}_{<j} = elements\_before\_j\_in\_order\_\pi$

For instance, we can imagine some ordering π on S where π is $a, b, c, d, e, q, z, j, y$. Then all the preceding elements would be included in the set $S^{\pi}_{<j} = \{a, b, c, d, e, q, z\}$, but not j itself.

Furthermore, we can also use the definition of submodularity combined with an ordering in to "build" a value function iteratively.

$V^{\pi}_j = V(S^{\pi}_{<j} + j) - V(S^{\pi}_{<j})$

Now we want to show that the following holds, which then proves out claim that all submodular functions are XOS.

Lemma 25.4 $V(S) = max_{\pi} \sum_{j \in S} V^{\pi}_j$

Proof: We can prove this by showing each direction of an inequality. Below we will define these as two lemmas. By lemma 25.4 and 25.5 we have proven the above. ■

Lemma 25.5 $V(S) \leq max_\pi \sum_{i \in S} V_i^\pi$

Proof: To show, this we can simply take the approach of constructing some π such that $V(S) = \sum_{i \in S} V_i^\pi$, and then the maximizer has to be at least this large.

In other words, we pick π_c s.t. $V(S) = \sum_{i \in S} V_i^{\pi_c}$

It turns out that picking any π_c such that the set S appears *first* in this ordering is enough. Assume S contains the first j elements in π_c. We use the previous definition:

$\sum_{i \in S} V_i^{\pi_c} = \sum_{i=1}^{j} V(S_{<i}^{\pi_c} + i) - V(S_{<i}^{\pi_c})$

Clearly, in the case where we've put $S = \{s_1, s_2, \ldots, s_j$ as the beginning of the ordering of π, then we get a nice telescoping sum where the terms cancel out such that the result is:

$= V(s_1) - V(\emptyset) + V(s_1, s_2) - V(s_1)) + ...V(S) - V(s_1, \ldots, s_{j-1})$
Since all items before the last item of S in the ordering π_c must be simply S, by construction. This leaves us with the following:
$= V(S) - V(\emptyset)$
which is clearly just
$= V(S)$ Thus we have shown the inequality one way.

■

Lemma 25.6 $V(S) \geq max_\pi \sum_{i \in S} V_i^\pi$

Proof: To show this is a little bit more complicated. Imagine you are given some arbitrary ordering π, and we want to show that the above holds. Start by doing the following - define π' such that it is the same as π except it has been shuffled such that S comes first in the ordering. For instance, if

$\pi = a, b, c, d, x, y, z$ and $S = x, y$

then

$\pi' = x, y, a, b, c, d, z$

Given the above, and the proof for the previous inequality, we thus know that:

$\sum_{i \in S} V_i^{\pi'} = V(S)$

This turns out to be very useful. If we can now show that

CLAIM: $V_i^\pi \leq V_i^{\pi'} \forall i \in S, \pi$

Then we are golden! Since the sum will then fulfill the inequality we are trying to prove. Now we can use a nice property of submodularity. We know that $S_{<i}^{\pi'} = (S_{<i}^{\pi} \cap S)$ for all $i \in S$, so $S_{<i}^{\pi'} \subseteq S_{<i}^{\pi}$, so by the decreasing marginal utility property we get the desired inequality. ■

25.3 Fractionally Subadditive = XOS

Since in the last lecture we proved that XOS $\subseteq$ Fractionally Subadditive we just need to prove that Fractionally Subadditive $\subseteq$ XOS to show the equivalence. Recall the definitions of Fractionally Subadditive:

Definition 25.7 (Fractionally Subadditive) *v is Fractionally Subadditive if for any $x_B \geq 0, \forall B$ whenever we have $\forall j \in A, \sum_{B|j \in B} x_B \geq 1$ this implies that we have $v(A) \leq \sum_B x_B v(B)$.*

25.3.1 Proof of containment

Lemma 25.8 *Fractionally Subadditive $\subseteq$ XOS*

Proof: In order to show this we will take Fractionally Subadditive valuations and show that there must exist corresponding XOS valuations. Specifically, we will show for any set A and a fractionally subadditive utility function v, there is a valuation v_j^A we can use that has $v(A) = \sum_{j \in A} v_j^A$.
When we say "a valuation we can use", we need that for any other set B, we have $v(B) \geq \sum_{j \in B} v_j^A$. This way, when we look at the maximum $\max_A \sum_{j \in B} v_j^A$ for some set B, this is occur in set $A = B$, and hence we will have $v(B) = \max_A \sum_{j \in B} v_j^A$ for all sets B, and hence have an XOS definition of v.

We will define v_j^A to be the usage of j as if we chose set A and note that we would like the following properties in our XOS function:

$$\begin{array}{ll} v(A) = \sum_{j \in A} v_j^A & \\ \sum_{j \in B} v_j^A \leq v(B) & \forall B \\ v_j^A \geq 0 & \forall j \in A \end{array} \tag{25.1}$$

We can view this is a classical linear program with variables $y_j = v_j^A$ and

$$\begin{array}{ll} max \sum_{j \in A} y_j & \\ \sum_{j \in B} y_j \leq v(B) & \forall B \\ y_j \geq 0 & \forall j \in A \end{array} \tag{25.2}$$

where the maximization is for the choice of y, and the question is, is this maximum as high as $v(A)$ (clearly cannot be higher as we have a constraint also for the set $B = A$.

Next we need a little review of linear programming to see what condition guarantees that the maximum is high enough.

25.3.2 Duality and Linear Programs

In order to show the correspondence of these functions we first need to understand duality as well as some nice properties that come with it: strong duality and weak duality. The idea of duality is that any constrained maximization problem has a corresponding constrained minimization problem in which we are trying to tighten the constraints as much as possible instead of trying to maximize in some direction subject to the constraints. In order to show this we will examine the following Linear Program with n non-negative decision variables y_i and m constraints (one for each subset). This can equivalently be written in matrix form as

$$\begin{array}{ll} \text{Maximize} & c^T y \\ \text{Subject To} & Ay \leq b \\ & y \geq 0 \end{array}$$

In which $Ay \leq b$ is saying that each row of Ay needs to be less than the corresponding element in b. So we have rows corresponding to sets B, with $b_B = v(B)$. The vector c has $c_j = 1$ for $j \in A$ and 0 otherwise.

Considering such a linear program more generally, we can multiply each constraint (corresponding to a set B) by a non-negative number, say x_B, and have x_B times the righthand side at most $x_B b_B$. Since $x_B \geq 0$, this keeps the direction of the inequality. Note that this corresponds to left multiplying the inequality $Ax \leq b$ on both sides by x^T so we have:

$$x^T A y \leq x^T b$$

If we can do this so that $x^T A \geq c^T$, then we get a bound for the maximum of the linear program above:

$$c^T y \leq x^T A y \leq x^T b$$

using that $y \geq 0$ also so we have multiply the $x^T A \geq c^T$ inequalities also with y. The dual linear program is the best bound one can get this way, that is, the smallest we can make the bound $x^T b$. Writing this out with transposing all, so it looks more like the above linear program, we get

$$\begin{array}{ll} \text{Minimize} & b^T x \\ \text{Subject To} & A^T x \geq c \\ & x \geq 0 \end{array} \tag{25.3}$$

By our derivation of the dual, we showed the following, which is called weak duality:

Lemma 25.9 (weak duality) *Any feasible solution to the dual 25.3 is an upper bound of any feasible solution to the primal 25.3.2*

The main theorem of linear programming (called string duality) states that if 25.3 and 25.3.2 both have a feasible solutions than the maximum and minimum are actually equal. We will not prove this theorem.

Theorem 25.10 (strong duality) *If there are solutions x and y to both 25.3 and 25.3.2, then the value of the maximum is equal to the minimum.*

Note that if the primal 25.3.2 is unbounded then there cannot exist and upper bound and the dual 25.3 has no feasible solution. Similarly, if the 25.3.2 is infeasible then we can push dual solution arbitrarily low and so the dual can be unbounded. It is also possible that they are both infeasible.

25.3.3 Using Linear programming duality.

Using the strong duality theorem, we can now show that Fractionally Subadditive functions are XOS.

Since we have that the function v is fractionally subadditive, we that for all $x_B \geq 0$ if $\sum_{B|j\in B} x_B \geq 1$ true $\forall j \in A$ then $\sum_B x_B v(B) \geq v(A)$. This can be summarized as:

$$\begin{array}{ll} v(A) \leq \min \sum_B x_B v(B) & \\ \sum_{B|j\in B} x_B \geq 1 & \forall j \in A \\ x_B \geq 0 & \forall B \end{array} \tag{25.4}$$

Now observe that the linear program 25.4 is exactly the dual of the linear program 25.2. Both clearly have some solution (for example $y_j = 0$ for all j, and $x_A = 1$ and $x_B = 0$ for all $B \neq A$). By the fractionally subadditive property we have that the minimum is at least $v(A)$, and by definition of the 25.2 we know that the maximum is at most $v(A)$ (as that is an upper bound). So we have max $\leq v(A) \leq$ min. By strong duality the two must be equal, and hence they must both equal to $v(A)$. This is what we wanted to prove.

■