

# GAME THEORY: ANSWERS TO HOMEWORK 2, BONUS QUESTION FROM HW 1

VISHVAS VASUKI

*Remark.* All external references point to [1].

## 1. 4.6 ON PAGE 101

Consider the algorithm described in Section 4.5 on page 93. Below, we use the same notation as the text. We want to improve the swap regret bound to  $O(\sqrt{NT \log N})$ .

We observe that the total loss experienced by the R external regret minimizing algorithms  $A_i$  is at most  $T$ . So, each  $A_i$  experiences a loss which is at most  $x_i T$ , such that:  $0 \leq x_i \leq 1$  and  $\sum_{i=1}^N x_i \leq 1$ .

Now consider theorem 4.6 described in page 87. It says:  $L_{PW}^T \leq L_k^T + \eta Q_k^T + \frac{\ln N}{\eta}$  for  $\eta \leq 1/2$ ,  $Q_k^T = \sum_{t=1}^T (l_k^t)^2$ . We apply this to the algorithm  $A_i$ . We observe:  $Q_k^T \leq \sum_{t=1}^T (l_k^t) \leq x_i T$ . We take  $\eta = \min \left\{ \sqrt{\frac{\ln N}{x_i T}}, 1/2 \right\}$ . Now, we get that external regret  $R_i$  for  $A_i$  is at most  $2\sqrt{x_i T \ln N}$ .

Consider equation 4.1 on page 93. For  $A_i$ , we have:  $\sum_{i=1}^T p_i^t \langle q_i^t, l^t \rangle \leq \sum_{t=1}^T p_i^t l_j^t + 2\sqrt{x_i T \ln N}$ .

As in section 4.5, we sum these up to get:  $L_H \leq L_{H,F} + 2 \sum_{i=1}^N \sqrt{x_i T \ln N}$ . But, by Jensen's inequality, as  $\sqrt{\cdot}$  is concave, we have  $\frac{\sum_{i=1}^N \sqrt{x_i}}{N} \leq \sqrt{\frac{\sum_{i=1}^N x_i}{N}} \leq \frac{1}{\sqrt{N}}$ .

Thus, we have:  $L_H \leq L_{H,F} + O(\sqrt{NT \log N})$ .

## 2. 5.1 ON PAGE 133

Consider the linear case of the Fisher model, where utilities are 0 or 1. The strongly polynomial algorithm will be identical to the algorithm described in Section 1.8 (pages 24, 25), except the process of finding a tight set during each iteration will be different. This process is described below.

Let  $x$  be the current uniform price,  $A$  the remaining set of goods,  $B$  the remaining set of buyers. Using  $x$ , construct the max-flow network as described in the book. Using the residual flow network, find the surplus money  $s_b$  for each buyer  $b$ . Now, for each good  $j \in A$ , compute  $s_j = \sum_{b \in \Gamma(j)} s_b$ , the total surplus money possessed by the buyers interested in good  $j$ . Now, uniformly increase the price of each good by adding  $\min_{j \in A} \frac{s_j}{a_j}$ , where  $a_j$  is the amount of the good  $j$ . Now,  $\left\{ \arg \min_{j \in A} \frac{s_j}{a_j} \right\}$  is a tight set.

The rest of the algorithm proceeds as usual.

The old algorithm was weakly polynomial only because it relied on binary search during each iteration in order to find the price  $x^*$  required to create a tight set. Thus, its running time depended on the numerical values of  $m_i, a_j$ . However, the

modification described above removes this dependency. Hence, we have a strongly polynomial algorithm.

*Remark.* Answer known to be incorrect.

### 3. 5.7 ON PAGE 134

We want to find simple algorithms for single source multiple sink markets where the underlying graph is a path or a tree. The algorithm needs to satisfy KKT conditions 1 and 3 mentioned in section 5.13 (page 125). Both the algorithms described below find equilibrium prices satisfying these conditions.

**3.1. The case of the path.** A sink, by definition, has no outgoing edges. So, here we have a path from a source to a sink. Let  $m$  be the money available at the sink. The algorithm simply finds the set of  $t$  edges (bottlenecks) in the path with the minimal capacity  $c$ , and sets the price of a unit flow in these edges to  $\frac{m}{tc}$ . The prices of all other edges are set to 0.

An alternative would be to simply set the price of a flow in one of the minimum capacity edges to be  $\frac{m}{c}$ .

**3.2. The case of the tree.** Without loss of generality we assume that the set of sinks is exactly the same as the set of leaves of the tree. Let  $c(e)$  denote the capacity of edge  $e$ . Let  $N(e)$  denote the (directed) edges whose tails are adjacent to the head of  $e$ . For each node  $v$ , define  $m(v)$  to be the total money possessed by all sinks which are in the subtree rooted at  $v$ .

We maintain a quantity called flow-capacity or  $fc(e)$  for each edge  $e$ . Doing a bottom-up traversal of the tree, we update  $fc$  for each edge  $e$  as follows: If  $e$  is adjacent to a leaf, set  $fc(e) = c(e)$ . Otherwise,  $fc(e) = \min(c(e), \sum_{e' \in N(e)} fc(e'))$ . If  $fc(e) = c(e)$ , the edge  $e = (u, v)$  is said to be a bottleneck for the subtree rooted at  $v$ .

For every sink / leaf  $t$ , we find the bottleneck edge  $e = (u, v)$  which is closest to the root. We then set the price of a flow in that edge to be  $\frac{m(v)}{c(e)}$ .

*Remark.* Answer known to be incorrect.

### 4. 6.1 ON PAGE 158: COBBS DOUGLAS DERIVATION ONLY.

Take equations in section 6.2 on page 141. We see that, when  $x_{i,j} \neq 0$ ,  $\lambda_{i,j} = 0$ . We use this in equation 6.4 to obtain  $\forall j$ :

$$\begin{aligned} \frac{e_i}{u_i(x_i)} \frac{\partial u_i(x_i)}{\partial x_{i,j}} &= \pi_j \\ \frac{e_i}{\prod_j x_{i,j}^{a_{i,j}}} \frac{a_{i,j} \prod_j x_{i,j}^{a_{i,j}}}{x_{i,j}} &= \pi_j \\ \frac{e_i a_{i,j}}{x_{i,j}} &= \pi_j \\ \frac{e_i a_{i,j}}{\pi_j} &= x_{i,j}(\pi) \end{aligned}$$

*Remark.* Comparison of answer to formula 6.7 on page 149 is done by setting  $\rho \rightarrow 0$ ,  $e_i = \sum_k \pi_k w_{i,k}$  and  $\sum_k \alpha_{ik} = 1$ . Indeed, the answer proves true.

5. 6.2 ON PAGE 158

Following the same reasoning as in page 149, we observe that equilibrium price is the set  $\pi \in R_{++}$  such that,  $\forall j: \sum_i x_{i,j}(\pi) \leq \sum_i w_{i,j}$ . Using the answer to the previous question, the constraints are of the form  $\sum_i \frac{e_i a_{i,j}}{\pi_j} \leq \sum_i w_{i,j}$ , where  $e_i = \sum_k \pi_k w_{i,k}$ . This may be rewritten as:  $\sum_i a_{i,j} \sum_k \pi_k w_{i,k} \leq \pi_j \sum_i w_{i,j}$ .

Clearly, both sides of the inequality are linear.

So, equilibrium price may be characterized as solutions to a linear feasibility program.

6. 1.6 ON PAGE 27 [BONUS QUESTION FROM HOMEWORK 1]

**6.1. Preliminaries.** Consider a two player game, between the column player c and the row player r. Take any pair of mixed strategies  $(P^c, P^r)$  constituting a Nash equilibrium for this game. Let the supports of these strategies be  $S^c$  and  $S^r$  respectively. Let their size be at most n. Let  $N'$  be the maximum utility which any player can obtain for any pure strategy pair.

Let M be the expected utility to c from the strategy  $P^c$  under this equilibrium. Let N be the analogous expected utility for r. Let  $u_r(s_r, s_c)$  be the utility for r when r plays the pure strategy  $s_r$  and c plays the pure strategy  $s_c$ . Let  $u_c(s_r, s_c)$  be the analogous utility for c.

**6.2. Building multisets.** Build a multiset  $\hat{S}^c$  of pure strategies by picking  $O(\log n/(\epsilon^2))$  independent samples from the distribution  $P^c$  over  $S^c$ . By an analogous process of sampling from the distribution  $P^r$ , build the multiset  $\hat{S}^r$ . Let  $|\hat{S}^r| = t$  and  $|\hat{S}^c| = v$ .

Let  $U^c$  and  $U^r$  represent the uniform distributions with the supports  $\hat{S}^c$  and  $\hat{S}^r$  respectively. We want to show that, with high probability, if c and r play  $U^c$  and  $U^r$ , they can expect utility close to M and N respectively.

**6.3. Deviation bounds used.** We now introduce deviation bounds which prove useful in our analysis.

Consider n iid random variables  $\{X_i\}$ , taking on values in  $[0, c]$ , with mean  $\mu$ . The Hoeffding bound  $Pr(|\frac{\sum_i X_i}{n} - \mu| \geq \epsilon c) \leq 2e^{-\frac{n\epsilon^2}{2}}$  is used. We call this the mean estimator bound.

**6.4. Comparing expected utilities of pure strategies relative to  $U^r$  and  $P^r$ .** First, we show that, for a fixed  $s_c$ ,  $u_c(s_c) = E_{s_r \sim P^r}[u_c(s_r, s_c)]$  is close to  $v_c(s_c) = E_{s_r \sim U^r}[u_c(s_r, s_c)] = \frac{\sum_{s_r \in \hat{S}^r} u_c(s_r, s_c)}{t}$  with probability  $\geq 1 - n^{-3}$  for large enough t. Using the mean estimator bound,  $Pr(|v_c(s_c) - u_c(s_c)| \geq \epsilon N') \leq 2e^{-t\epsilon^2} = n^{-3}$  for some  $t = O(\log n/(\epsilon^2))$ .

Applying the union bound, we conclude that, with probability  $\geq 1 - tn^{-3} \geq 1 - n^{-2}$ , for every  $s_c \in \hat{S}^c$ ,  $|u_c(s_c) - v_c(s_c)| \leq \epsilon N'$ . So,  $|\sum_{s_c \in \hat{S}^c} u_c(s_c) - \sum_{s_c \in \hat{S}^c} v_c(s_c)| \leq \epsilon n N'$  with high probability. This means that  $|\frac{\sum_{s_c \in \hat{S}^c} u_c(s_c)}{v} - \frac{\sum_{s_c \in \hat{S}^c} v_c(s_c)}{v}| \leq N'\epsilon$  with high probability.

**6.5. Comparing expected utility of  $U^c$  and M.** We now show that the expected utility of  $U^c$  is close to M with high probability.

We note, using the mean estimator bound, that  $Pr(|\frac{\sum_{s_c \in \hat{S}_c} u_c(s_c)}{v} - M| \geq \epsilon N') \leq 2e^{-v\epsilon^2/2}$ . So, for some  $v = O(\log n/(\epsilon^2))$ , we can conclude that  $|\frac{\sum_{s_c \in \hat{S}_c} u_c(s_c)}{v} - M| \leq \epsilon N'$  with probability  $1 - n^{-2}$ .

Using results shown earlier, and applying the union bound, we can say that, with probability  $1 - O(n^{-2})$ ,  $|M - \frac{\sum_{s_c \in \hat{S}_c} v_c(s_c)}{v}| \leq 2N'\epsilon$ .

**6.6. Concluding remarks.** Therefore, given any  $\epsilon' = 2(M + N' + N)\epsilon$ , we can construct  $(U^c, U^r)$  with  $O(\log n/(\epsilon^2))$  sized supports, such that, with high probability, the expected utility of  $(U^c, U^r)$  for  $c$  is at most  $\epsilon' M$  away from  $M$ .

Similar arguments as those presented above will show that  $r$  will achieve a utility at most  $\epsilon' N$  away from  $N$  with high probability.

Applying the union bound, we see that, with high probability, both  $c$  and  $r$  achieve utility close to  $M$  and  $N$  respectively.

**6.7. An  $\epsilon$  approximate Nash equilibrium?** I do not yet have a clear and complete argument about why  $(U^c, U^r)$  would be an  $\epsilon$  approximate Nash equilibrium. Intuition tells me that this is not the case in general.

#### REFERENCES

- [1] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.