

15780: GRADUATE AI (SPRING 2017)

Homework 4: Game Theory and Social Choice  
(Solutions)

Release: April 20, 2017,  
Due: May 4, 2017

# 1 Swap Regret and Correlated Equilibrium [25 points]

A *modification function*  $F_i : S_i \rightarrow S_i$  for each player  $i$  is a function that maps each action in  $S_i$  to an action in  $S_i$ , where  $S_i$  is the strategy space of player  $i$ . Given a modification function  $F_i$  for player  $i$ , we define the *swap regret* of player  $i$  with respect to  $F_i$  as follows:

$$\text{regret}_i(s, F_i) = u_i(F_i(s_i), s_{-i}) - u_i(s_i, s_{-i}).$$

That is,  $\text{regret}_i(s, F_i)$  measures the regret player  $i$  has for playing  $s_i$  rather than  $F_i(s_i)$ , in the strategy profile  $s$ .

For simplicity, consider a two-player game where  $N = \{1, 2\}$ . A distribution  $p$  over  $S_1 \times S_2$  is an  $\epsilon$ -*correlated equilibrium* if and only if the following conditions are met:

- $\sum_{s_1 \in S_1} \left( \max_{s'_1 \in S_1} \left( \sum_{s_2 \in S_2} p(s_1, s_2) u_1(s'_1, s_2) - \sum_{s_2 \in S_2} p(s_1, s_2) u_1(s_1, s_2) \right) \right) \leq \epsilon.$
- $\sum_{s_2 \in S_2} \left( \max_{s'_2 \in S_2} \left( \sum_{s_1 \in S_1} p(s_1, s_2) u_2(s_1, s'_2) - \sum_{s_1 \in S_1} p(s_1, s_2) u_2(s_1, s_2) \right) \right) \leq \epsilon.$

Assume players 1 and 2 played  $T$  games and each player followed strategies with swap regret at most  $R$ . That is, for each player and *every possible* modification function, the sum of regrets over  $T$  games was at most  $R$ . Let  $q$  be the empirical distribution of the joint actions played by the players. That is, if we let  $(s_1^t, s_2^t) \in S_1 \times S_2$  be the strategy profile at each  $t$ -th game, then for each strategy profile  $(s_1, s_2) \in S_1 \times S_2$ ,  $q(s_1, s_2) = \frac{1}{T} |\{t \in [T] : (s_1, s_2) = (s_1^t, s_2^t)\}|$ .

Prove that the distribution  $q$  over  $S_1 \times S_2$  is an  $(R/T)$ -correlated equilibrium.

**Hint:** The definition of an  $\epsilon$ -correlated equilibrium can be rewritten using swap regret.

**Solution:** The definition of an  $\epsilon$ -correlated equilibrium can be rewritten using the swap regret. For a distribution  $p$  over  $S_1 \times S_2$ , let  $F_1^{(p)}$  be a modification function that maps each  $s_1 \in S_1$  to  $\bar{s}_1 \in S_1$  such that

$$\sum_{s_2 \in S_2} p(s_1, s_2) u_1(\bar{s}_1, s_2) = \max_{s'_1 \in S_1} \sum_{s_2 \in S_2} p(s_1, s_2) u_1(s'_1, s_2).$$

Likewise, let  $F_2^{(p)}$  be a modification function that maps each  $s_2 \in S_2$  to  $\bar{s}_2 \in S_2$  such that

$$\sum_{s_1 \in S_1} p(s_1, s_2) u_2(s_1, \bar{s}_2) = \max_{s'_2 \in S_2} \sum_{s_1 \in S_1} p(s_1, s_2) u_2(s_1, s'_2).$$

Then,  $p$  is an  $\epsilon$ -*correlated equilibrium* if and only if the following conditions are met:

- $\sum_{s_1 \in S_1} \left( \sum_{s_2 \in S_2} p(s_1, s_2) \times \text{regret}_1((s_1, s_2), F_1^{(p)}) \right) \leq \epsilon.$
- $\sum_{s_2 \in S_2} \left( \sum_{s_1 \in S_1} p(s_1, s_2) \times \text{regret}_2((s_1, s_2), F_2^{(p)}) \right) \leq \epsilon.$

Now we prove that  $q$  is a  $(R/T)$ -correlated equilibrium by showing that  $q$  satisfies the two conditions with  $\epsilon = R/T$  in the modified definition.

$$\sum_{s_1 \in S_1} \left( \sum_{s_2 \in S_2} q(s_1, s_2) \times \text{regret}_1((s_1, s_2), F_1^{(q)}) \right)$$

$$\begin{aligned}
&= \sum_{s_1 \in S_1} \sum_{s_2 \in S_2} (1/T) |\{t \in [T] : (s_1, s_2) = (s_1^t, s_2^t)\}| \times \text{regret}_1((s_1, s_2), F_1^{(q)}) \\
&= \sum_{s_1 \in S_1} \sum_{s_2 \in S_2} \sum_{t \in [T] : (s_1, s_2) = (s_1^t, s_2^t)} (1/T) \times \text{regret}_1((s_1, s_2), F_1^{(q)}) \\
&= \sum_{s_1 \in S_1} \sum_{s_2 \in S_2} \sum_{t \in [T] : (s_1, s_2) = (s_1^t, s_2^t)} (1/T) \times \text{regret}_1((s_1^t, s_2^t), F_1^{(q)}) \\
&= \sum_{t=1}^T (1/T) \times \text{regret}_1((s_1^t, s_2^t), F_1^{(q)}) = (1/T) \times \sum_{t=1}^T \text{regret}_1((s_1^t, s_2^t), F_1^{(q)}) \leq R/T.
\end{aligned}$$

Likewise,

$$\sum_{s_2 \in S_2} \left( \sum_{s_1 \in S_1} q(s_1, s_2) \times \text{regret}_2((s_1, s_2), F_2^{(q)}) \right) \leq R/T.$$

Thus, the distribution  $q$  is a  $(R/T)$ -correlated equilibrium.

## 2 Repeated Insertion Model and the Mallows Model [25 points]

Prove that the Mallows Model with parameter  $\phi$  is equivalent to the Repeated Insertion Model (RIM) with  $p_{ij} = \phi^{i-j} \frac{1-\phi}{1-\phi^i}$ , where  $p_{ij}$  is the probability of the  $i^{\text{th}}$  element being inserted in the  $j^{\text{th}}$  spot, for  $i \geq j$ .

**Hints:** As a first step, note that the RIM is equivalent to generating a vector  $v$  of  $m$  elements, where the  $i^{\text{th}}$  element is an integer in  $[1, i]$  corresponding to the location at which the  $i^{\text{th}}$  ranked alternative in the true ranking is inserted in the current subranking. It may be useful to first consider the relationship between the insertion vector and the Kendall tau distance between the resulting ranking and the true ranking. In addition, you may use the fact that given a true ranking  $\succ$  over  $m$  alternatives and the space of all rankings over the same alternatives  $\Pi$ ,  $(1 + \phi)(1 + \phi + \phi^2) \cdots (1 + \phi + \cdots + \phi^{m-1}) = \sum_{\succ' \in \Pi} \phi^{d_{KT}(\succ, \succ')}$ .

**Solution:** First, note that the RIM is equivalent to generating a vector  $v$  of  $n$  elements, where the  $i^{\text{th}}$  element is an integer in  $[1, i]$  corresponding to the location at which it is inserted in the current subranking. It is clear that this vector uniquely defines a complete ordering over all alternatives. Furthermore, given a reference ranking  $\sigma$ , the vector  $v$  that corresponds to the correct final ranking is  $\{1, 2, \dots, n\}$ .

Now, given a vector  $v$  that generates a final ranking  $\rho$ , the Kendall-Tau distance between  $\sigma$  and  $\rho$  is the L1 norm between  $v$  and  $\{1, 2, \dots, n\}$ . That is,  $d_{KT}(\sigma, \rho) = \sum_{i=1}^n (i - v_i)$ . This is because when you put the  $i^{\text{th}}$  item in spot  $j < i$ , this necessarily flips  $i - j$  comparisons between item  $i$  and other items that are supposed to be ranked before it. The argument then proceeds by induction on the number of elements.

The probability of any ranking  $\rho \in \Pi$  can be decomposed as follows:

$$\begin{aligned}
Pr[\pi] &= p_{1,v(1)} p_{2,v(2)} \cdots p_{n,v(n)} \\
&= \phi^{1-v(1)} \frac{1-\phi}{1-\phi^1} \cdot \phi^{2-v(2)} \frac{1-\phi}{1-\phi^2} \cdots \phi^{n-v(n)} \frac{1-\phi}{1-\phi^n} \\
&= \left( \phi^{1-v(1)} \phi^{2-v(2)} \cdots \phi^{n-v(n)} \right) \frac{(1-\phi)^n}{\prod_{k=1}^n (1-\phi^k)} \\
&= \phi^{d_{KT}(\rho, \sigma)} \frac{(1-\phi)^n}{\prod_{k=1}^n (1-\phi^k)} \quad (\text{because of the above fact}) \\
&= \phi^{d_{KT}(\rho, \sigma)} \left( \frac{1-\phi}{1-\phi} \cdot \frac{1-\phi}{1-\phi^2} \cdots \frac{1-\phi}{1-\phi^n} \right)
\end{aligned}$$

$$\begin{aligned}
&= \phi^{d_{KT}(\rho, \sigma)} \left( 1 \cdot \frac{1}{1 + \phi} \cdots \frac{1}{1 + \phi + \cdots + \phi^{n-1}} \right) \\
&= \phi^{d_{KT}(\rho, \sigma)} \frac{1}{\sum_{\tau \in \Pi} \phi^{d_{KT}(\tau, \sigma)}}. \qquad \text{(by the hint)}
\end{aligned}$$

This is the same as the Mallows model, so by setting  $p_{ij} = \phi^{i-j} \frac{1-\phi}{1-\phi^i}$  in RIM, we get the Mallows  $\phi$ -model.

### 3 Strategyproof Social Choice on a Tree [25 points]

Recall that we informally proved in class that given the single peaked preferences of a set of voters on a line, selecting the median of the points is both strategyproof and Condorcet consistent. In this problem, you will design a social choice function for a slightly more general setting that is both strategyproof and Condorcet consistent.

A city wants to build a new library, and they want to elicit the residents' preferences to determine where to build it. The locations that the city is able to build the library can be represented by a set of vertices  $V$  which lie on a tree  $T = (V, E)$  (for example, the root of the tree might be located in downtown while the leaves are on the fringes of the city).

- There are an odd number of residents (voters), indicated by the set  $N$ .
- Each voter  $i \in N$  has a most preferred vertex  $v_i \in V$ .
- Let  $d(x, y)$  denote the length (in terms of number of edges) of the unique path between vertices  $x$  and  $y$  in  $T$ . Given two vertices  $u$  and  $w$ , if  $d(v_i, u) < d(v_i, w)$ , then voter  $i$  prefers  $u$  to  $w$ .

Write a social choice mechanism (i.e., a function) that takes as input the most preferred vertex of each voter and outputs a single vertex  $v \in V$ . Your proposed mechanism should be strategyproof and Condorcet consistent. Formally prove that these properties are satisfied.

**Solution:** There are several valid mechanisms, which can be shown to be equivalent. For example one mechanism would be to output the vertex

$$v = \arg \min_{v' \in V} \sum_{i \in N} d(v_i, v')$$

Notice that this mechanism is a generalization of the median to the tree (see [https://en.wikipedia.org/wiki/Geometric\\_median](https://en.wikipedia.org/wiki/Geometric_median)).

Another (equivalent) mechanism is to arbitrarily root the tree, and output the vertex  $v$  that is furthest from the root where the subtree rooted at  $v$  contains more than  $|N|/2$  of the votes. Note that such a vertex has to exist since the whole tree contains all  $|N|$  votes, so if no vertex below the root has a subtree that contains more than  $|N|/2$  votes, then we would simply output the root. Furthermore, note that such a node is unique, because for there to be two or more vertices that are equidistant from the root and whose subtrees contain more than  $|N|/2$  votes each, then the subtree that contains all of those vertices must contain more than  $|N|$  votes.

We will show that the latter mechanism is strategyproof and Condorcet consistent.

*Strategyproof:* Assume, by way of contradiction, that a voter  $i$  could manipulate the election. This voter's preferred vertex  $v_i$  must either be contained in the subtree rooted at  $v$  or outside of this subtree.

- If  $v_i$  is in the subtree rooted at  $v$ , if voter  $i$  still votes in this subtree, then notice that  $v_i$  must be contained in a subtree rooted at one of the children of  $v$ . In that case, voter  $i$  would only benefit by making the mechanism output a vertex in that subtree; however, in order to do so it must stay in that subtree, which would mean the subtree would need to contain more than  $|N|/2$  votes to begin with, which cannot be the case by design of the mechanism. If voter  $i$  changes their vote to be outside the subtree rooted at  $v$ , it could only potentially cause the mechanism to change its output to some vertex  $w$  that is outside of the subtree. But  $d(v_i, w) > d(v_i, v)$ , so  $v_i$  has no incentive to change their vote.
- If  $v_i$  is outside of the subtree rooted at  $v$ , then voting outside the subtree could not impact the outcome of the mechanism, and voting inside the subtree could only cause the mechanism to change its output to some vertex  $u$  that is inside the subtree. But  $d(v_i, u) > d(v_i, w)$ , so  $v_i$  has no incentive to change their vote.

Thus we arrive at a contradiction, proving that the mechanism is strategyproof.

*Condorcet Consistent:* To prove that the mechanism is Condorcet consistent, we have to show that in any two candidate election where  $v$  is one of the candidates,  $v$  will get the majority of the votes. There are two cases.

- If the other candidate is in the subtree rooted at  $v$ , then it must be contained in a subtree rooted at one of the children of  $v$ . Any subtree rooted at one of the children of  $v$  will contain less than  $|N|/2$  preferred vertices of voters by design of the mechanism (and since  $|N|$  is odd), so since  $v$  would get all the votes from preferred vertices outside of that subtree,  $v$  would win.
- If the other candidate is outside of the subtree rooted at  $v$ , then  $v$  would clearly win, because it would get at least all of the votes from preferred vertices contained in its subtree, which is more than  $|N|/2$  by design of the mechanism.

## 4 Programming: Stackelberg Strategies [25 points]

In a 2-player normal form game, a Stackelberg strategy is where one of the players is a leader and the other is a follower. In contrast to the default situation where both players pick their respective strategies at the same time, a Stackelberg strategy is when the leader, which is identified as player 1, first commits to a (mixed) strategy which the follower, player 2, knows. Then player 2 commits to his own strategy using his knowledge of player 1's strategy.

An optimal Stackelberg strategy would be a Stackelberg strategy where player 1's expected utility is maximized. The optimal Stackelberg strategy can be computed in polynomial time by solving multiple LPs. See Slide 14 of Lecture 19 for a description of the algorithm.

Given a 2-player normal form game, you will implement the function `stackelberg(u1, u2)` in `stackelberg.py` which will return the optimal Stackelberg strategy for the given game. You can use `cvxopt` or `cvxpy` to solve the LPs. Your function should return numpy arrays, not datatypes from these libraries. If there is a tie in the expected utility between two strategies of player one (i.e. the expected utility is off by an absolute error of  $1e-5$ ), you should return the optimal strategy induced by the lowest indexed pure strategy of player 2.

## 5 Submitting to Autolab

Create a tar file containing your writeup and the completed `stackelberg.py` modules for the programming problems. Make sure that your tar has these files at the root and not in a subdirectory. Use the following commands from a directory with your files to create a `handin.tgz` file for submission.

```
$ ls
stackelberg.py  writeup.pdf  [...]
$ tar cvzf handin.tgz writeup.pdf stackelberg.py
a writeup.pdf
a stackelberg.py
$ ls
handin.tgz  stackelberg.py  writeup.pdf  [...]
```