Final

Steven An, A12921886
CSE254, Freund

March 15, 2020

Problem 1

For convenience, let the outcomes and predictions be -1, and 1 instead of 0 and 1. As the set of experts is divisible, we may use the potential function presented in the slides on drifting games. From those slides, the minmax optimal strategy of the adversary is to split each of the k bins evenly, which is always possible due to experts being divisible. The minmax optimal strategy for the master on iteration r is to choose

$$b(r) = \text{sign}\left(\sum_{i=1}^{k} d(i,r)2^{1-r}\binom{r-1}{i}\right)$$

where $d(i,r)$ is the difference in size of the adversary's split in bin i on iteration r. That is, if the probability mass in bin k is n, then there are two ways of splitting it into two sets, e.g. $0.4n$ and $0.6n$. If the split is below $0.5n$, then the value of $d(\cdot,\cdot)$ is negative. The potential function is

$$\Psi(r) = \Psi(r-1) + b(r)\sum_{i=1}^{k} d(i,r)2^{1-r}\binom{r-1}{i}.$$

From the slides, we know adversary's goal is to maximize this while the master's is to minimize it. Since it must be the case that in the optimal strategy, no experts are wrong when the master does not make a mistake, we can consider iterations where the master makes a mistake.

To see that the strategy laid out is minmax, it suffices to consider the potential function. Notice that the sum by definition (of the master's prediction) will always be negative. Further, the definition of the potential function implies it is negative. This means that if any $d(i,r)$ is non-zero, then the adversary can improve by making it zero. Hence, the adversary must play as described. Since the master is only allowed to predict 1 or -1, the only other choice for the master is to predict with the opposite sign. This means that if the recurrence is unrolled, there is a sum which is positive (because the master chose to play something else) when it could have been negative. This means that the master could improve by changing their prediction for that iteration. Hence, the master must play as described. Thus, the strategy laid out is minmax.

Problem 2

Part (a)

We use the standard method of finding Nash equilibria to find the minmax equilibria. First, we may note that there are no pure minmax equilibria because for any pure strategy from either player, there is a response by the other player which is such that the first player can improve by changing their choice. Now, let p be the probability that the row player plays the first row. Similarly, let q be the probability the column player plays the first column. It suffices to solve $-2p + 1 - p = p - 1 + p$ and $2q - 1 + q = -q + 1 - q$. We have $p = 0.4$ and $q = 0.4$. Then, both players play the first row/column respectively with probability 0.4. The minmax value is 0.2.

Part (b)

Since Nash equilibria are correlated equilibria, we may notice that there are three Nash equilibria. First is for either player to play either column/row respectively with probability 0.5. Then, the other two are where the row player plays the first row, the column player the second column and where the row player plays the second row and the column player the first column. The first Nash equilibrium is easy to check using the standard method. The other two are apparent from the game structure. The only other correlated equilibria is where an impartial agent draws cards representing the latter two Nash equilibria with any probability. This is because the Nash equilibria that have pure strategies give the same payout. The mixed Nash equilibrium gives value 0. The other Nash equilibria and the correlated equilibria give value 1.

Problem 3

As there are k bits in the seed, there are 2^k possible seeds. Hence, as we are assumed to know the procedure behind the pseudo random number generator (PRNG), we record the results of the PRNG for each of the seeds. Then, regardless of the seed input into the PRNG, the resulting binary sequence must be one of the 2^k sequences observed. Refer now to each of the 2^k outcomes as experts and the sequence that we must predict as the 'sequence'. We can predict the sequence in an online fashion, using the experts for advice. The halving algorithm may be used because we know that one of the expert's predictions will always be correct. Using the bound for the number of mistakes the predictor will have using the halving algorithm, we make at most k mistakes when predicting the sequence, as required.

Problem 4

Part (a)

One may represent each of the symbols using 7 bits as $\log_2 128 = 7$. As the message length is T, one only needs $7T$ bits to encode it.

Part (b)

In expectation, arithmetic encoding achieves a compressed sequence of approximately $TH(\hat{P})$ bits, $\hat{P}$ the empirical distribution of the characters. From the slides we know that if there is a distribution p over all possible messages of length T, then the expected code length is $1 + H(p_T)$ where

$$H(p_T) = -\sum_{(x_1,x_2,\ldots,x_T)} p(x_1, x_2, \ldots, x_T) \log_2 p(x_1, x_2, \ldots, x_T).$$

Further, it is also known that the per character entropy, $H(P)$, is $\lim_{T\to\infty} H(p_T)/T$. It is the case then that $H(p_T)$ is an approximation of $TH(P)$. By the law of large numbers, the empirical distribution characters will converge to the truth distribution of characters. So, $H(\hat{P})$ is an approximation of $H(P)$ and so $H(p_T)$ approximates $TH(\hat{P})$. Hence, we have an approximate upper bound on the expected number of bits needed to encode the message using arithmetic encoding at $1 + TH(\hat{P})$. Recalling that Shannon's lower bound is $(1 - o(1))TH(P)$, we can use our approximation to see that it is about $(1 - o(1))TH(\hat{P})$, meaning the upper bound is tight.

To actually do the encoding, we first need to partition the unit interval. For every character in the message, partition the each interval (from the current partition) into 128 segments, where the i^{th} segment has length equal to the empirical probability of the i^{th} character. That is, for the first character, partition the unit interval into 128 segments. For the second, partition each of those 128 intervals into 128 intervals, etc. To retrieve the encoding, we must find the interval that represents the message. From the previous procedure, we have T sets of intervals, each a refinement of the previous. If the first character is the i^{th} character, we restrict our attention to the i^{th} interval in the first of our T intervals. If the second character is the j^{th} character, we restrict our attention to the j^{th} interval inside the i^{th} interval from before. This is done for the remaining $T-2$ characters. Then, the result will be an interval, and it suffices to take any binary number that falls in that interval.

PROBLEM 5 The outcomes are $x_1, x_2, \ldots, x_T$, expert predictions at time t are $z_{t,1}, z_{t,2}, \ldots, z_{t,N}$ for the N experts and the master's prediction is z_t^M.

PART (A)

We present the *EG* algorithm given by Kivinen and Warmuth in "Additive Versus Exponentiated Gradient Updates for Linear Prediction" (1995). However, the loss bound can be improved if the $EG^{\pm}$ algorithm is used, and the latter's bound is also presented.

Parameters: Prior distribution $\mathbf{p}_1$; experts numbering N; learning rate η; number of trials T.
Algorithm EG
Do for $t = 1, 2, \ldots, T$:

1. Predict with:

$$z_t^M = \frac{\sum_{i=1}^N p_{t,i} z_{t,i}}{\sum_{i=1}^N p_{t,i}}.$$

2. Observe outcome x_t and incur loss $(z_t^M - x_t)^2$.

3. Calculate a new posterior distribution:

$$p_{t+1,i} = \frac{p_{t,i} e^{-2\eta z_{t,i}(z_t^M - x_t)}}{\sum_{j=1}^N p_{t,j} e^{-2\eta z_{t,i}(z_t^M - x_t)}}.$$

On page 214, Lemma 4 states that the total loss of the exponentiated gradient algorithm presented above has loss less than or equal to

$$\left(1 + \frac{c}{2}\right) \sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{z}_t - x_t)^2 + \left(\frac{1}{2} + \frac{1}{c}\right) R^2 \mathbf{RE}(\mathbf{u}||\mathbf{p}_1)$$

where c is an arbitrary constant and R is upper bounds the maximum difference for expert predictions for every iteration.

The $EG^{\pm}$ algorithm is an improvement on the above which allows the weights become negative (by representing them as positive numbers). Kivinen and Warmuth present two different bounds, depending on how much information the master has. In the first claim of Theorem 2 on page 213, the loss of the modified exponentiated gradients algorithm is

$$3 \left(\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{z}_t - x_t)^2 + \|u\|_1^2 \left(\max_{i=1,2,\ldots N} ||\mathbf{z}_t||_\infty \right)^2 \ln 2N \right).$$

If we know the total loss using weight vector $\mathbf{u}$ is bounded above by K, then with an appropriate learning rate, the loss bound is

$$\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{z}_t - x_t)^2 + 2\|u\|_1 \left(\max_{i=1,2,\ldots N} ||\mathbf{z}_t||_\infty \right) \sqrt{2K \ln 2N} + 2\|u\|_1^2 \left(\max_{i=1,2,\ldots N} ||\mathbf{z}_t||_\infty \right)^2 \ln 2N.$$

Part (b)

A modified version of the exponentiated gradient algorithm found in "Using and combining predictors that specialize" by Freund et. al. (1997) is presented along with its loss bound.

From page 6, we have the **SEG** algorithm.

Parameters: Prior distribution $\mathbf{p}_1$; learning rate η; number of trials T.
Algorithm SEG
Do for $t = 1, 2, \ldots, T$:

1. Predict with:

$$z_t^M = \frac{\sum_{i \in E_t} p_{t,i} z_{t,i}}{\sum_{i \in E_t} p_{t,i}}.$$

2. Observe outcome x_t and incur loss $(z_t^M - x_t)^2$.

3. Calculate a new posterior distribution: if $i \in E_t$

$$p_{t+1,i} = p_{t,i} e^{-2\eta z_{t,i}(z_t^M - x_t)} \frac{\sum_{j \in E_t} p_{t,j}}{\sum_{j \in E_t} p_{t,j} e^{-2\eta z_{t,i}(z_t^M - x_t)}}.$$

Otherwise $p_{t+1,i} = p_{t,i}$

Here, we update the weights assigned to experts based on whether they were awake at the previous iteration. Further, the total weight of the awake experts does not change because of the numerator in the update rule.

It is derived on the next page that the loss of **SEG** given prior distribution $\mathbf{p}_1$ to any distribution $\mathbf{u}$ is

$$\sum_{t=1}^{T} u(E_t)(z_t^M - x_t)^2 \leq \frac{2}{2-\eta} \sum_{t=1}^{T} u(E_t)(\mathbf{u}^{E_t} \cdot \mathbf{z}_t^{E_t} - x_t)^2 + \frac{1}{\eta} \mathbf{RE}(\mathbf{u}||\mathbf{p}_1).$$

In comparison to the first bound given for the vanilla exponentiated gradient algorithm, this bound improves the coefficients of the relative entropy term. Assuming the first term is similar in both bounds, we have less loss associated with the relative entropy term.