Statistical Techniques in Robotics
Assignment 1 - Luka Eerens

2. Theory Questions:

2.1 Regret (5 Points)
Can regret be negative? If yes, provide an example. If no, provide a mathematical explanation. Note: Your answer depends on the assumptions you make. Make sure to state them with your answer.

Answer:
In short yes! If you assume that all experts can make mistakes including the best ones than it is possible to accumulate negative regret. Below is a case that shows this (you need at least 3 expert to for this). We are also assuming that each expert does not get eliminated after a mistake:

Round	Expert1	Expert2	Expert3	Learner	**Ground Truth**
1	1	1	0	1	1
2	1	0	1	1	1
3	0	1	1	1	1

In this kind of scenario, the learner would in fact not have made any errors, whereas all of the experts would have and since regret is a relative term, (the difference between learner loss and expert loss), the magnitude of the expert loss would be greater here, because they made mistakes. As a result under circumstances like these, regret can be negative.

2.2 Construct a hypothesis class H to make this a realizable scenario. In other words, show that in the case where we have a finite input space and a finite output space, we can always construct a hypothesis class (a set of experts), which contains the perfect expert, no matter the true labeling for the feature vectors. What is the size of such a hypothesis class?

Answer:
From the preamble to the question, the input space and output space will be stated below as they are needed to explain this:

Input Space: $X = \{x_1, x_2, \dots, x_N\}$

Output Space $Y = \{1, 2, \dots, K\}$

If we assume independent experts, to be absolutely right in output space, you need K experts, and so to be right in all rounds (n) , you need K^n experts.

Below is a concrete example proving this:
Number of features N = 3, number of classes K = 2 (ie: 0,1)

	Feature 1	Feature 2	Feature 3
Possibility 1	0	0	0
Possibility 2	0	0	1
Possibility 3	0	1	0
Possibility 4	1	0	0
Possibility 5	1	1	0
Possibility 6	1	0	1
Possibility 7	0	1	1
Possibility 8	1	1	1

This is thus 2^3 so K^n

2.3 Understanding Penalty Parameter η (10 Points)

2.3.1 In each case derive the optimal value of η given N and m^*. Consider if there are any constraints you can put on η.

Answer:
The optimal value of η occurs when the expected reward $E[R]$ is bounded. We thus need to find the inflection point of the derivative of expected reward with respect to η.

Thus

$$\frac{dE[R]}{d\eta} = 0$$

Now, the regret bound for the random weighted majority algorithm is also given by:

$$E[R] = \eta m^* + \frac{lnN}{\eta}$$

Deriving this expression with respect to η gives:

$$\frac{dE[R]}{d\eta} = m^* - \frac{lnN}{\eta^2}$$

Thus:

$$0 = m^* - \frac{lnN}{\eta^2}$$

Solving for η gives:

$$\eta = \pm\sqrt{\frac{lnN}{m^*}}$$

It needs to positive so that you bound it o < η <=1/2 but still need so finally

$$\eta = \sqrt{\frac{lnN}{m^*}}, \qquad \eta > 0$$

So with the bound in mind, we get:

$$\eta = +\min\left(\sqrt{\frac{lnN}{m^*}}, \qquad \frac{1}{2}\right)$$

Now, the regret bound for the weighted majority algorithm (not randomized) is:

$$E[R] = (1 + 2\eta)m^* + 2\frac{lnN}{\eta}$$

Deriving this expression with respect to η gives:

$$\frac{dE[R]}{d\eta} = 2m^* - 2\frac{lnN}{\eta^2}$$

Thus:

$$0 = 2\left(m^* - \frac{lnN}{\eta^2}\right)$$

Solving for η gives:

$$\eta = \pm\sqrt{\frac{lnN}{m^*}}$$

It needs to positive so that you bound it o < η <=1/2 but still need so finally

$$\eta = \sqrt{\frac{lnN}{m^*}}, \qquad \eta > 0$$

So with the bound in mind, we get:

$$\eta = +\min\left(\sqrt{\frac{lnN}{m^*}}, \qquad \frac{1}{2}\right)$$

This is the same as randomized.

2.3.2 Suppose we don't know m^* at the time we choose η (this is usually the case). However, suppose we know that the number of prediction rounds T is much smaller than N. In each case, should we choose a bigger or smaller value for η?

Answer:
It is better for η to be bigger.

If you have much fewer prediction rounds than the number of experts, you will need to more aggressively prune the number of experts from the list, and so will need a larger penalty on those experts that are wrong.

Whatever large value η is, it cannot be larger than ½, otherwise you mindlessly prune experts too quickly that would in reality give the best advice.

One other thing, that has to be considered, is that if you reach a situation, where each expert gives similar advice, the algorithm will NOT learn anything with each cycle of expert advice, as the dynamics of weight updates to each expert remains homogenous for all.

2.3.3 In each case, can we pick η to make the algorithm no regret for the following cases of m^*? If yes, provide η such that the algorithm is no-regret.
Answer:

a) $m^* = O(T)$

WMA
If m^* is linear, η should be such that E[R] becomes sub-linear in T in order for E[R] to be a no regret algorithm. Now:
If $\eta = \frac{1}{T}$ then complexity becomes as follows:

$$E[R] \leq (1 + 2\eta)m^* + \frac{lnN}{\eta}$$

$$E[R] \leq m^* + 2\eta m^* + \frac{lnN}{\eta}$$

$$E[R] \leq O(T) + \left(\frac{1}{T}\right) O(T) + O(T)$$

$$E[R] \leq O(T) + const + O(T)$$

This is linear E[R] which means that it is not possible to have no regret for WMA with $\eta = \frac{1}{T}$ if $m^* = O(T)$.
Now if $\eta = \frac{1}{\sqrt{T}}$ then complexity becomes as follows:

$$E[R] \leq O(T) + \left(\frac{1}{\sqrt{T}}\right) O(T) + O(\sqrt{T})$$

$$E[R] \leq O(T) + O\left(\frac{1}{\sqrt{T}}\right) + O(\sqrt{T})$$

This is not sub-linear E[R] which means that it is not possible to have no regret for WMA with $\eta = \frac{1}{\sqrt{T}}$ if $m^* = O(T)$.

b) $m^* = O(T^x)$, x < 1, ie., m^* is sub-linear in T.

If $\eta = \frac{1}{T}$ then complexity becomes as follows:

$$E[R] \leq m^* + 2\eta m^* + \frac{lnN}{\eta}$$

$$E[R] \leq O\left(\frac{1}{\sqrt{T}}\right) + \left(\frac{1}{T}\right) O\left(\frac{1}{\sqrt{T}}\right) + O(T)$$

$$E[R] \leq O\left(\frac{1}{\sqrt{T}}\right) + \left(\frac{1}{T\sqrt{T}}\right) + O(T)$$

This is not sub-linear E[R] which means that it is not possible to have no regret for WMA with $\eta = \frac{1}{T}$ and where $m^* = O(T^x)$, x < 1, ie., m^* is sub-linear in T.
Now if $\eta = \frac{1}{\sqrt{T}}$ then complexity becomes as follows:

$$E[R] \leq O(\frac{1}{\sqrt{T}}) + \left(\frac{1}{\sqrt{T}}\right) O(\frac{1}{\sqrt{T}}) + O(\sqrt{T})$$

$$E[R] \leq O\left(\frac{1}{\sqrt{T}}\right) + O\left(\frac{1}{T}\right) + O(\sqrt{T})$$

In this case, $E[R]$ is sub-linear as all sub-terms are sub-linear. Therefore in the case of this weighted majority algorithm, it is possible to achieve no-regret under the following conditions:

$$\eta = \frac{1}{\sqrt{T}}, \quad m^* = O(T^x), \quad x < 1, \quad \text{ie., } m^* \text{ is sub-linear in T}$$

RWMA

a) $m^* = O(T)$

If m^* is linear, η should be such that E[R] becomes sub-linear in T. It should thus be:

$$\eta = \frac{1}{\sqrt{T}}$$

If $\eta = \frac{1}{T}$ then complexity becomes as follows:

$$E[R] \leq \eta m^* + \frac{lnN}{\eta}$$

$$E[R] \leq O\left(\frac{1}{T}\right) O(T) + O(T)$$

$$E[R] \leq const + O(T)$$

Here E[R] is linear if $\eta = \frac{1}{T}$ and if $m^* = O(T)$ so these conditions forbid no-regret.

Whereas if $\eta = \frac{1}{\sqrt{T}}$ then complexity becomes as follows:

$$E[R] \leq \eta m^* + \frac{lnN}{\eta}$$

$$E[R] \leq O\left(\frac{1}{\sqrt{T}}\right) O(T) + O(\sqrt{T})$$

$$E[R] \leq O(\sqrt{T}) + O(\sqrt{T})$$

Here E[R] is sub-linear if $\eta = \frac{1}{\sqrt{T}}$ and if $m^* = O(T)$.

b) $m^* = O(T^x)$, x < 1, ie., m^* is sub-linear in T.

If m^* is sub-linear, η should be such that E[R] becomes sub-linear in T. It should thus be:

Here if $\eta = \frac{1}{T}$ you obtain the following:

$$E[R] \leq O\left(\frac{1}{T}\right) O(\sqrt{T}) + O(T)$$

$$E[R] \leq O\left(\frac{1}{\sqrt{T}}\right) + O(T)$$

Which is obviously not sub-linear as $O(T)$ is linear. Instead we must make η a constant independent of T. Doing so gives us:

$$E[R] \leq O(1) * O(\sqrt{T}) + O(1)$$

Here E[R] is sub-linear.

2.4.1
Show that for the fully observable scenario, the Halving algorithm has the same mistake bound as the binary label case. Here, the prediction at every round t is the label … Blablabla.

Answer:
In the beginning (at timestep t=1) there are E experts such that:

$$|V^1| \leq E$$

After the first iteration, you are bound to have at least 50% of experts get eliminated and so after M number of mistakes, t steps later you should have the quantity of experts denoted by the following expression:

$$|V^1| \leq \left(\frac{1}{2}\right)^M E$$

This represents the upper bound of the version space.

The lower bound of the version space is $1 \leq |V^1|$. Combining them together we obtain:

$$1 \leq |V^1| \leq \left(\frac{1}{2}\right)^M E$$

So comparing the lower and upper bound together yields the following expression:

$$1 \leq \left(\frac{1}{2}\right)^M E$$

Computing the log of both sides to find the value of M gives:

$$0 \leq -M + log_2 E$$

$$M \leq log_2 E$$

2.4.2 Preamble… Convert the above high-level strategy into pseudo-code for an algorithm and derive its mistake bound.

Answer:

$\boldsymbol{V}^{(1)} = H$
$\boldsymbol{for}\ t = 1, \ldots, T\ \boldsymbol{do}$
  $Receive\ (\boldsymbol{x}^{(t)})$
  $\hat{y}^t = MAJORITYCONSENSUS(\boldsymbol{V}^t, \boldsymbol{x}^{(t)})$
  $RECEIVE(y^t)$
  $if\ y^t\ != \hat{y}^t$:
    $\boldsymbol{V}^{(t+1)} = \{h \in \boldsymbol{V}^{(t)}: h(x^{(t)}) = y^{(t)}\}$

Derivation:
If there are K possible outputs and one correct ground truth answer, then with each cycle of expert advice, there must be at least 1 out of K experts making a mistake with each cycle. Also if we assume realizibility, the lower mistake bound for number of experts left at the end is at least 1. Therefore:

$$1 \leq |\boldsymbol{V}^t|$$

And as mentioned before, after M number of mistakes, at a rate of 1/k mistakes per turn, you get:

$$|\boldsymbol{V}^t| \leq \left(\frac{1}{k}\right)^M E$$

Thus

$$1 \leq |\boldsymbol{V}^t| \leq \left(\frac{1}{k}\right)^M E$$

and so:

$$1 \leq \left(\frac{1}{k}\right)^M E$$

If the number of outputs is 2 (binary) and we log both sides, we return to the original value reached in 2.4.1:

$$M \leq log_2 E$$

So as a general purpose if the number of outputs is k, we get:

$$M \leq log_k E$$

2.5

2.5.1. Show that the strategy exists for the adversary such that the loss (the number of mistakes) is always maximized for the deterministic Weighted Majority Algorithm. You should be able to explain the strategy in a few sentences. Hint: Consider a simple case with just 2 experts.

Answer:

An adversarial output has access to weights for each expert, as well as their advices. In the case of a deterministic weighted majority, the adversary would just go against the combined prediction of the experts and since the experts are confronted themselves with deterministic data, and so the adversary would go against data that can be determined. This intuitively should lead to a maximized loss. What better way then to output the opposite of what can be observed in order to maximize loss. An example of this in action would be experts A and B having weights of trustworthiness of 0.5, and 0.4 respectively, and the adversary choosing to output the prediction of expert B no matter what. So if expert A gave label 1, and expert B gave a 0, it would choose to output 0 despite expert A having a better track record.

2.5.2. Assume that at least one expert is correct at least once. Prove that the expected loss of the Randomized Weighted Majority Algorithm against the worst adversary (as designed above) is strictly less than the loss of WMA. Why does randomization help improve the worst, case performance of the learner?

Answer:

The worst adversary wrt WMA is when there is a 50:50 split among experts.

For RWMA the prediction does not come from majority vote, but instead comes from probabilities that are weighted.

If we define w_i as the weight of the ith expert, then it follows that the probability of picking them out of all of the expert will be expressed as:

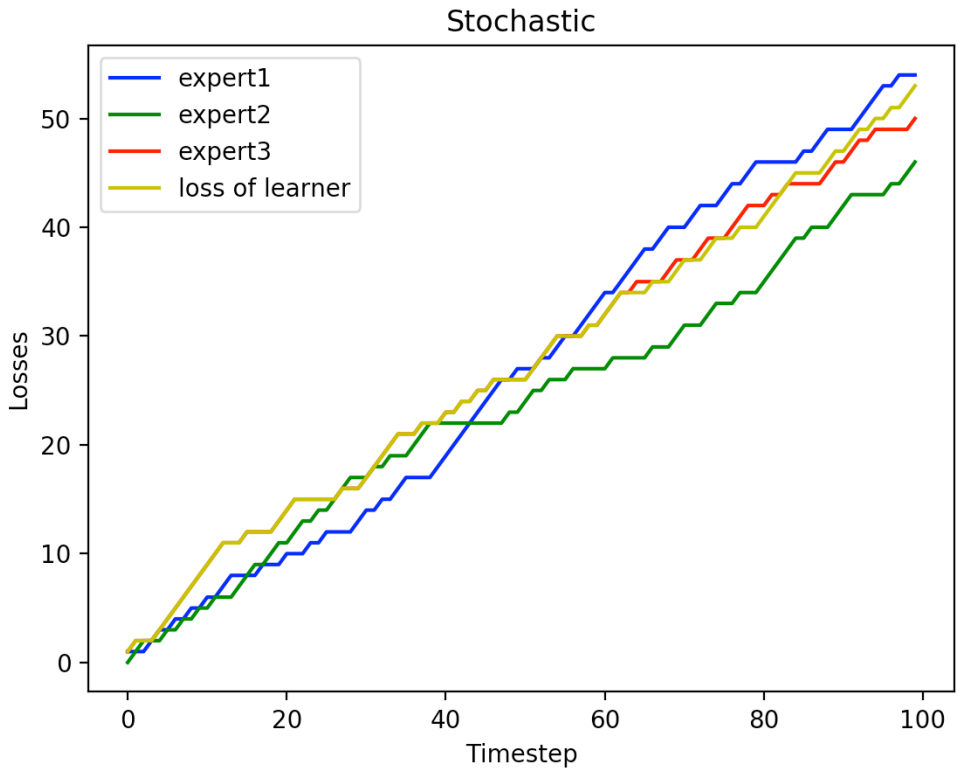
$$P(Learner\ Mistake) = \frac{\max(w_i, \dots, w_n)}{\sum_i^n (w_i, \dots, w_n)}$$

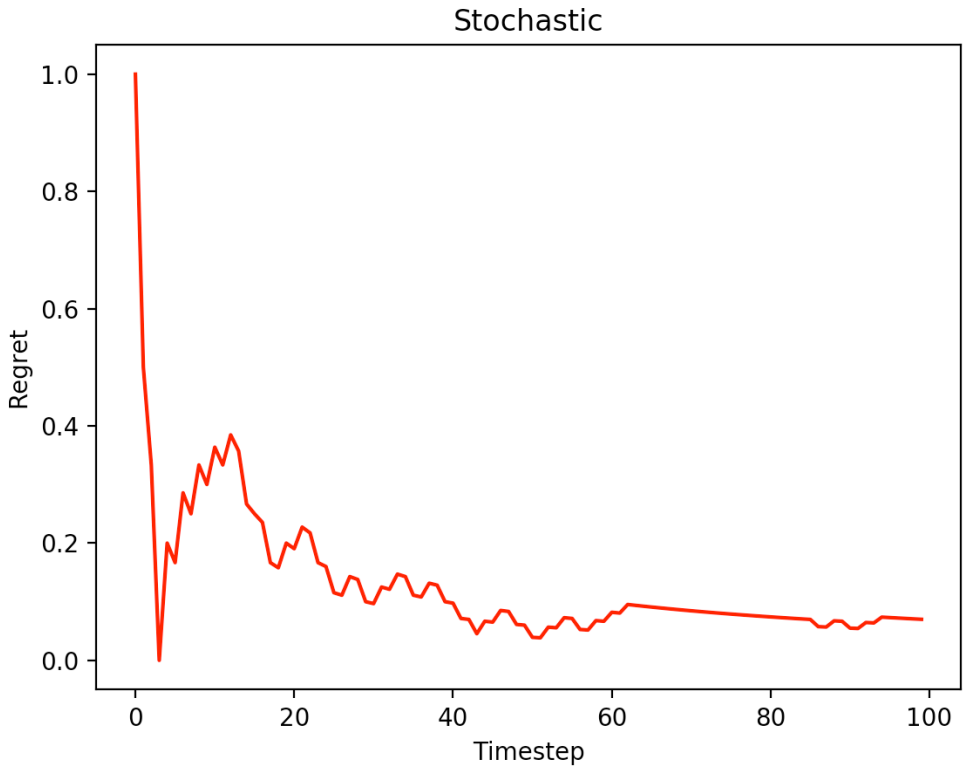
This fraction, which is smaller than 1 is then multiplied at each timestep to the number of mistakes done so far by the learner. This is not done directly with WMA, as there the number of mistakes M is just incremented as time t passes, whereas here with RWMA you basically have this weight being multiplied to M with each increment such that $P(Learner\ Mistake) * M$.

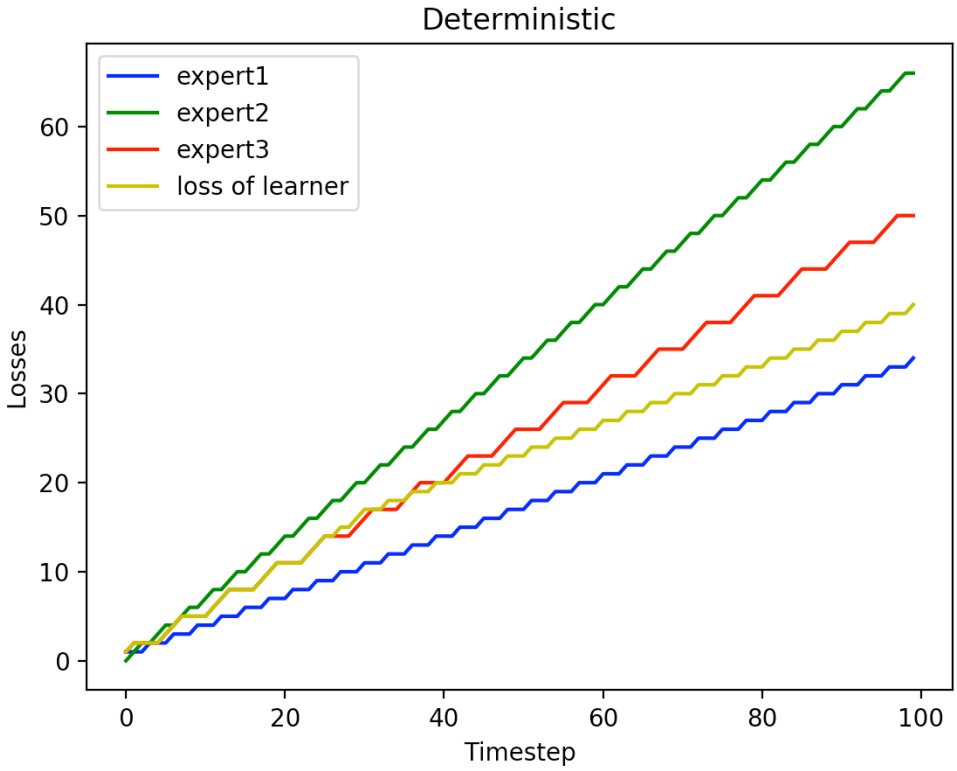
Also $P(Learner\ Mistake) < 1$ so with each timestep the number of mistakes M is smaller than with pure WMA.

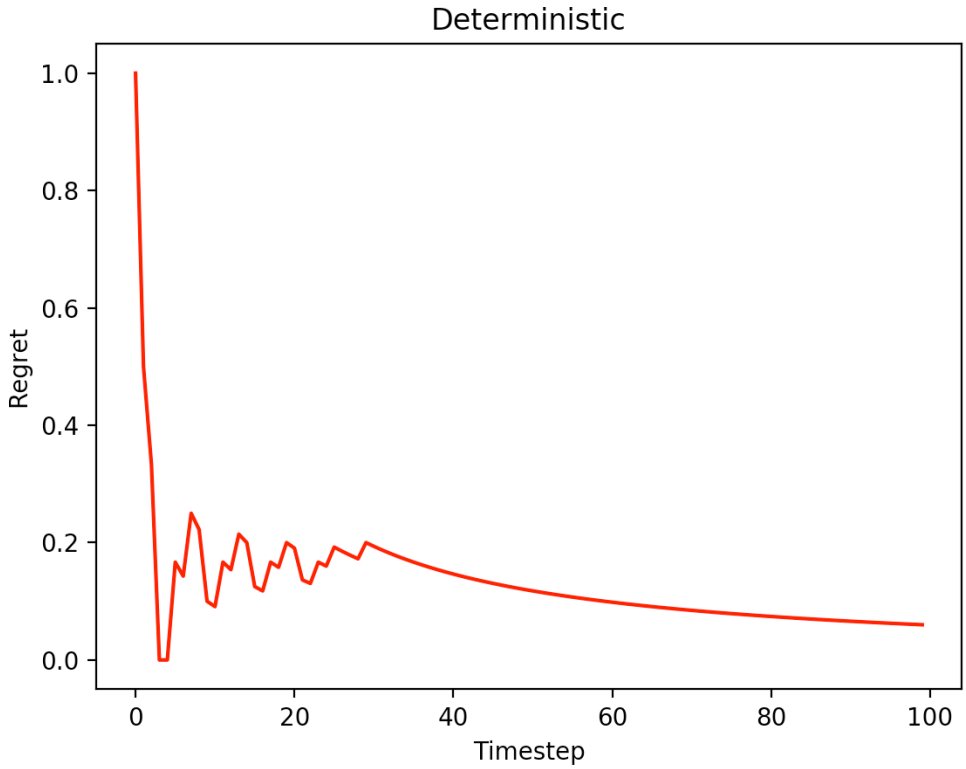
Randomization helps improve the worst-case performance of the learner because the learner behaves stochastically instead of deterministically under randomization. For adversarial outputs, the loss is less influenced by the proclivity for nature to produce adverse outputs. When the decision on which experts to consider becomes randomized the like RWMA under these circumstances, the result is worse loss for the worst case performance of the learner, and so better performance.

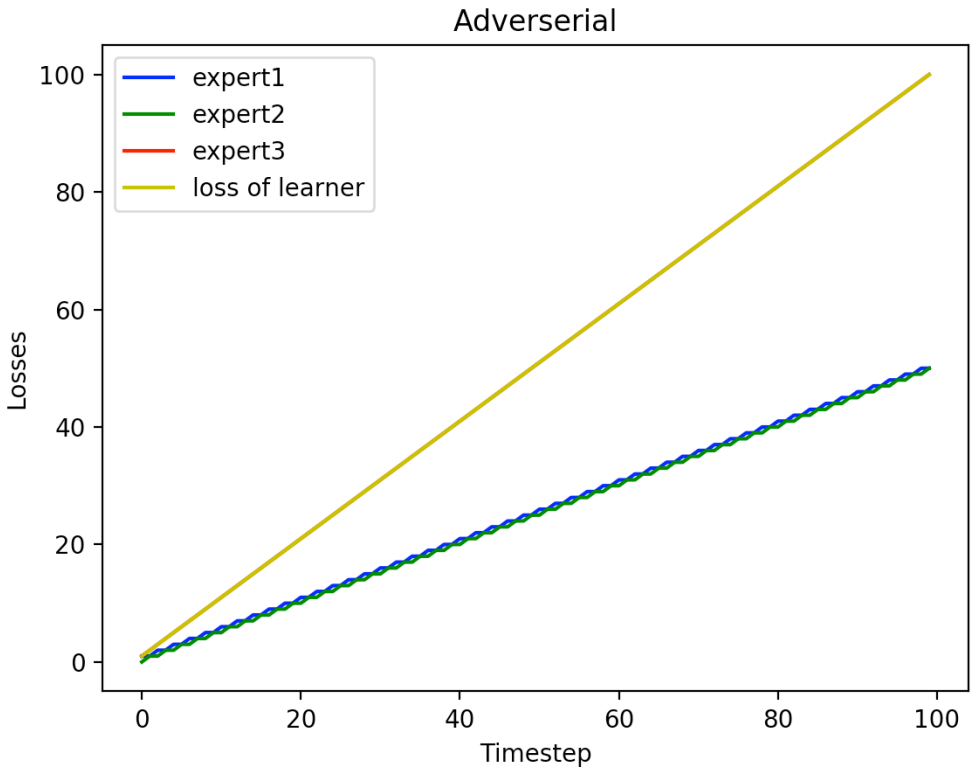
3.3 Implement the Weighted Majority Algorithm (15 Points)

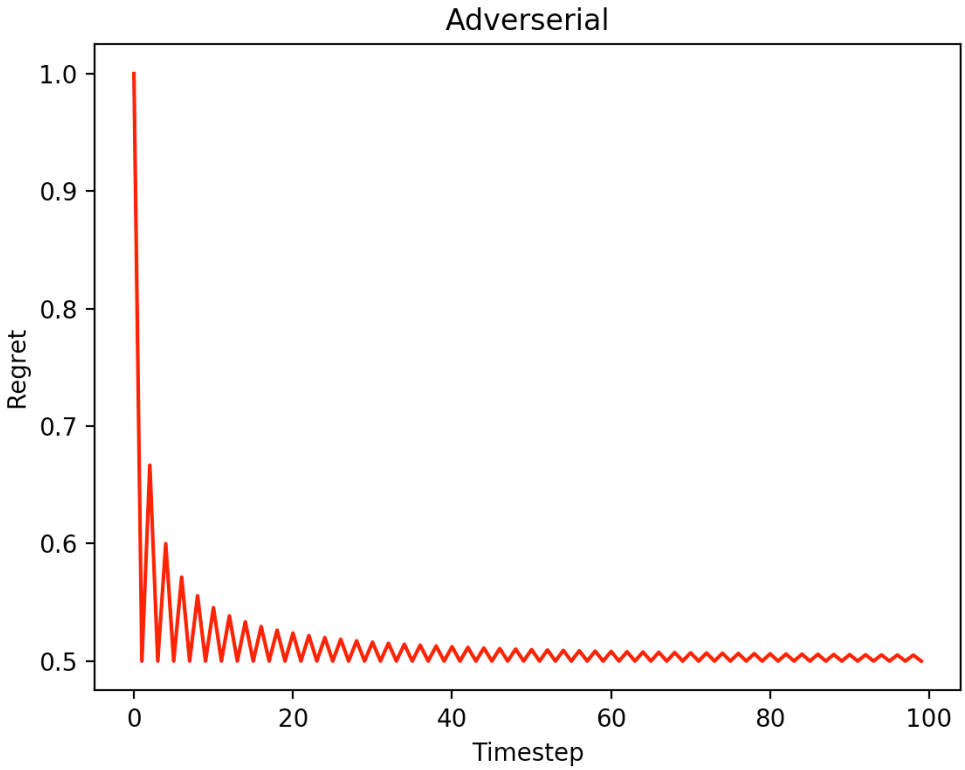





Deterministic
expert1
expert2
expert3
loss of learner
Losses
Timestep
0
10
20
30
40
50
60
0
20
40
60
80
100

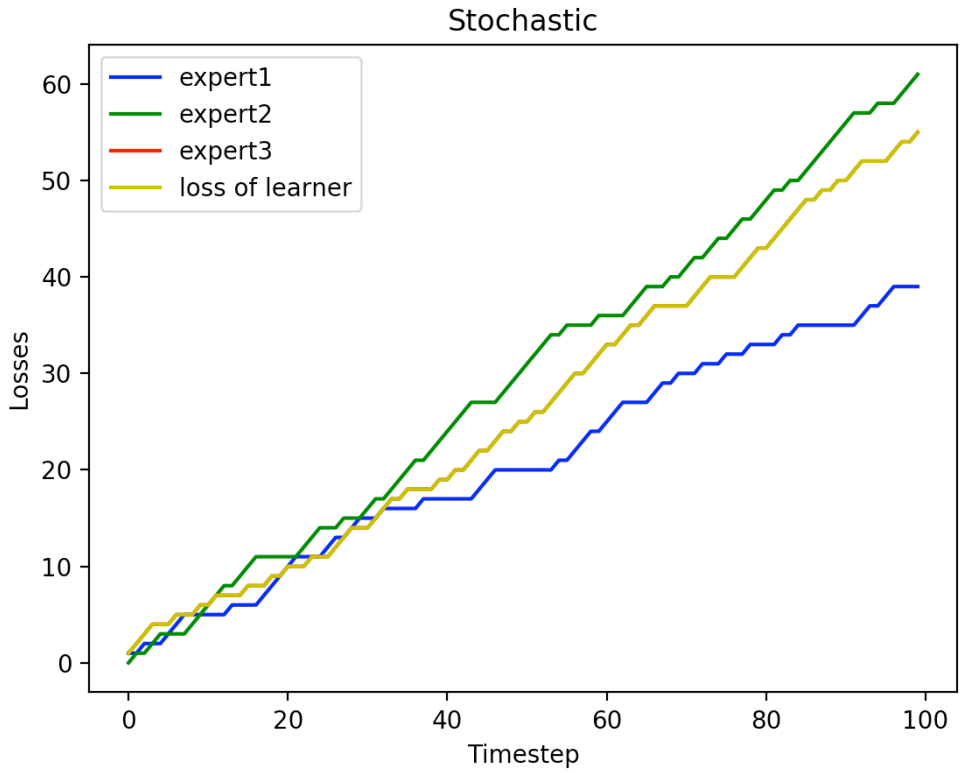

Deterministic
Regret
Timestep
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100

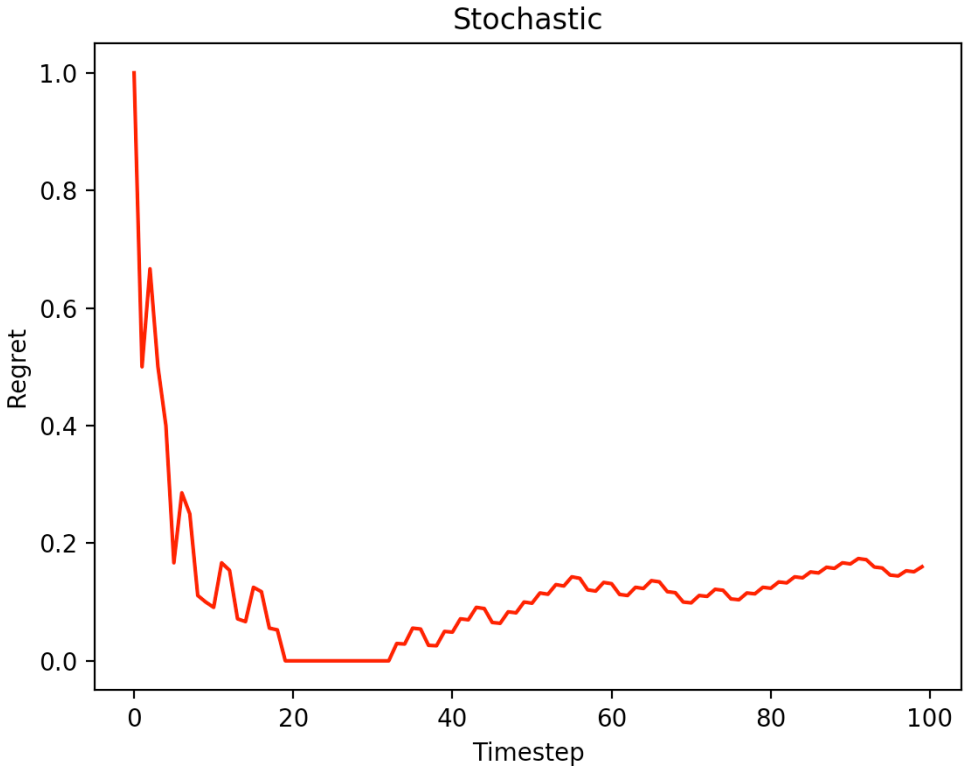

Adverserial
expert1
expert2
expert3
loss of learner
Losses
Timestep
0
20
40
60
80
100


Adverserial
Regret
Timestep
0.5
0.6
0.7
0.8
0.9
1.0
0
20
40
60
80
100

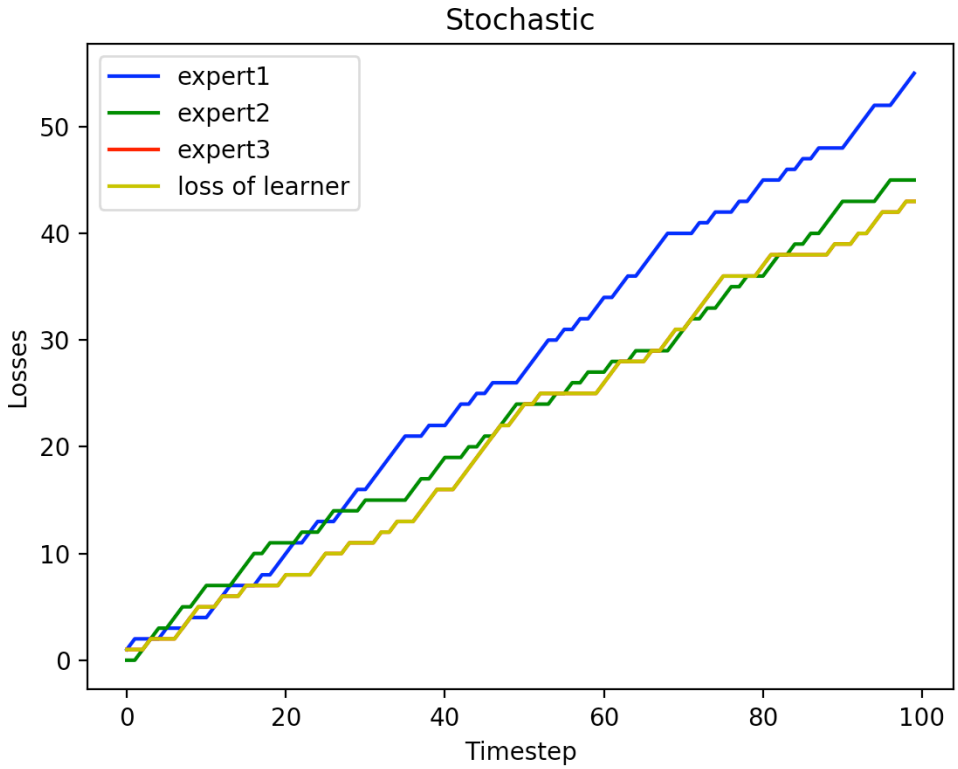
3.4 Implement the randomized weighted majority algorithm (15 points)

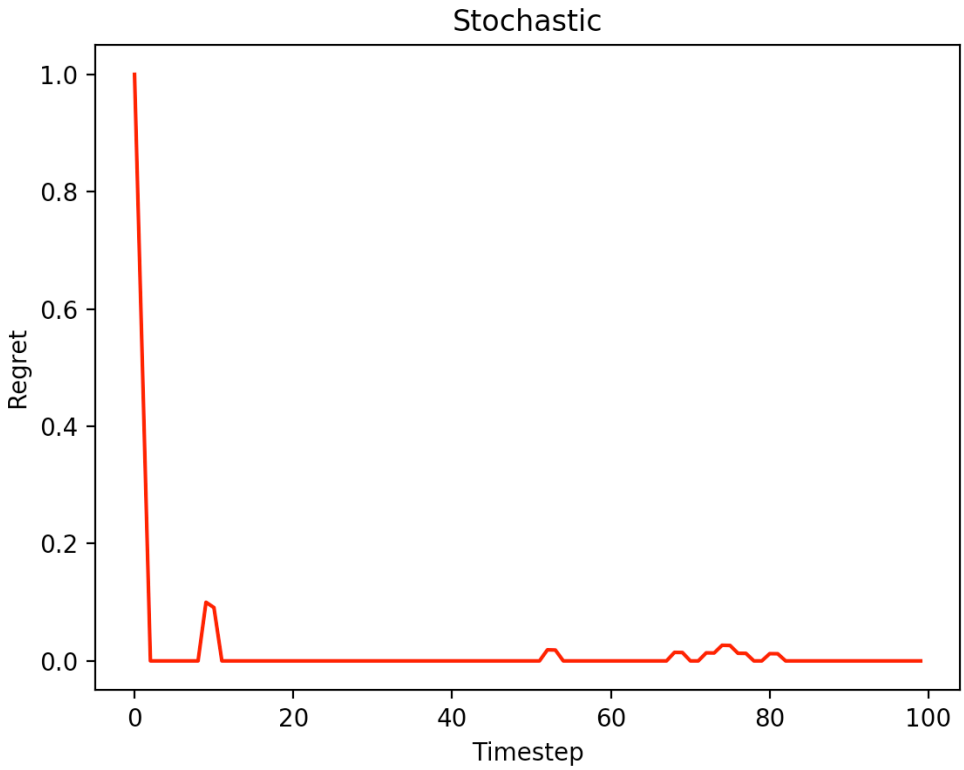
The first thing we will do is retain the same value of $\eta = \frac{1}{2}$



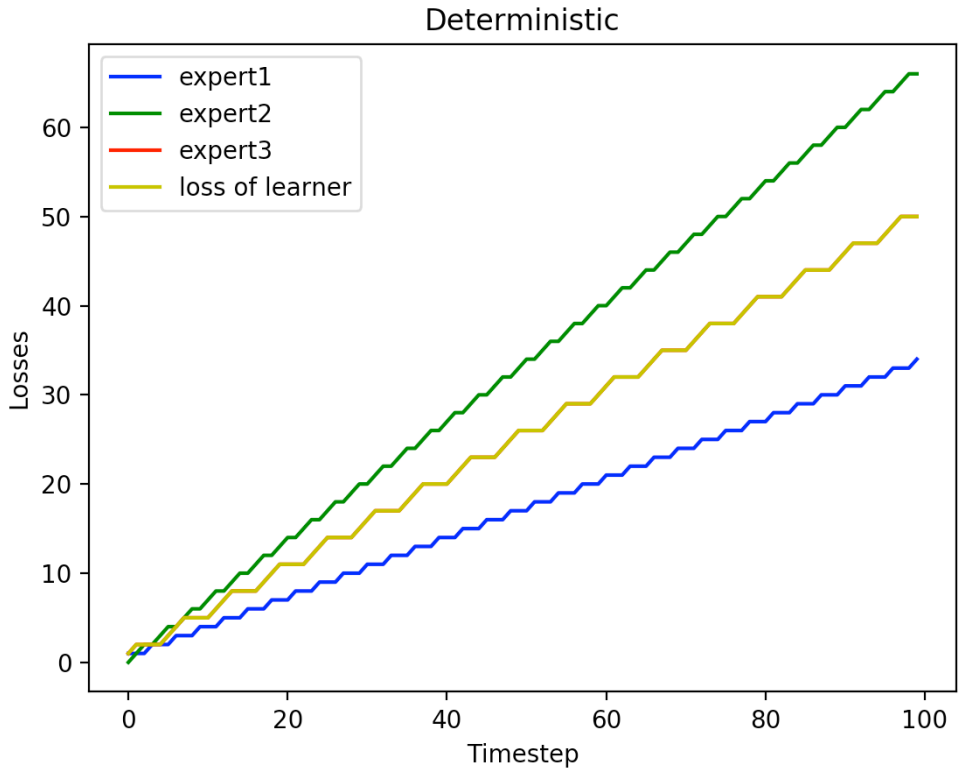


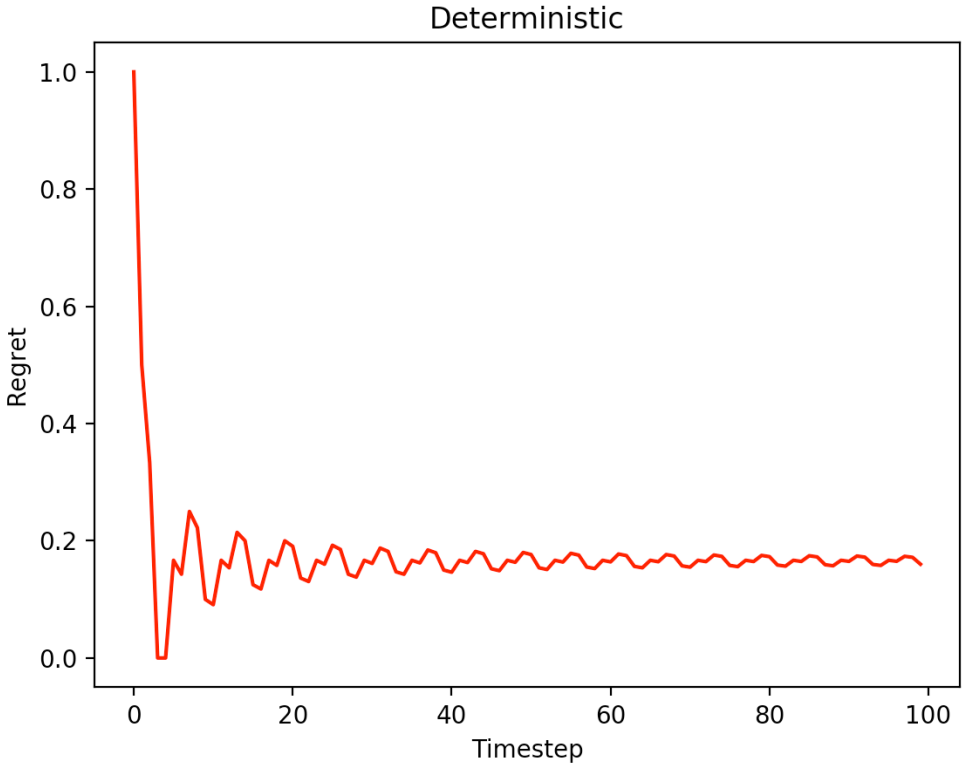
Staying on stochastic we will now change to $\eta = \frac{1}{10}$



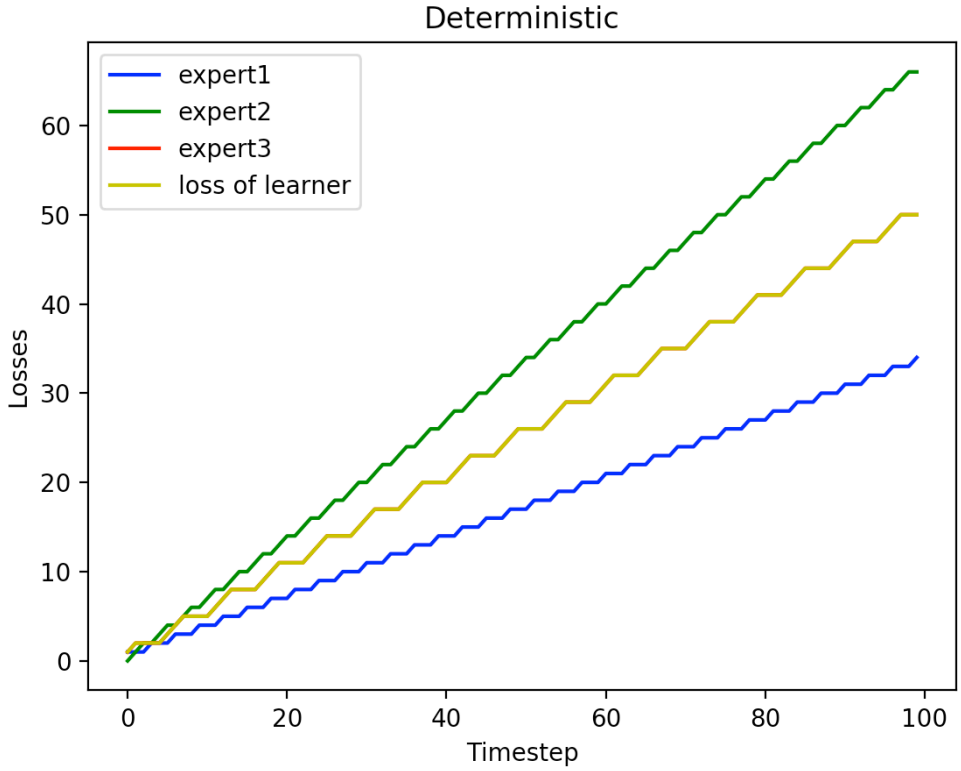


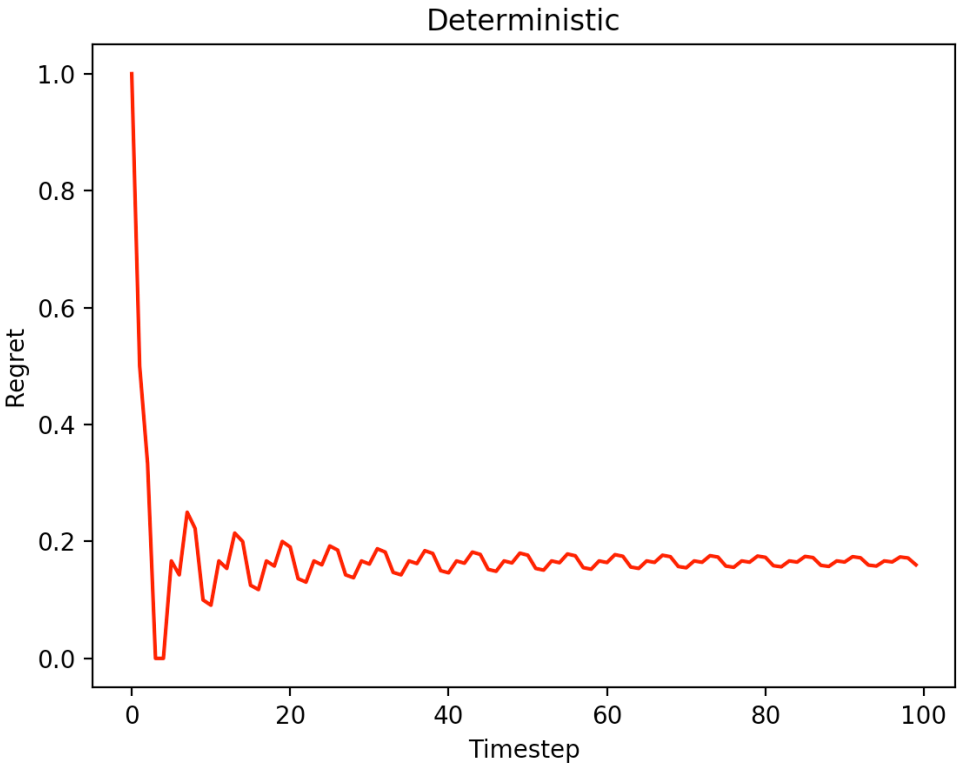
Now onto deterministic at $\eta = \frac{1}{2}$



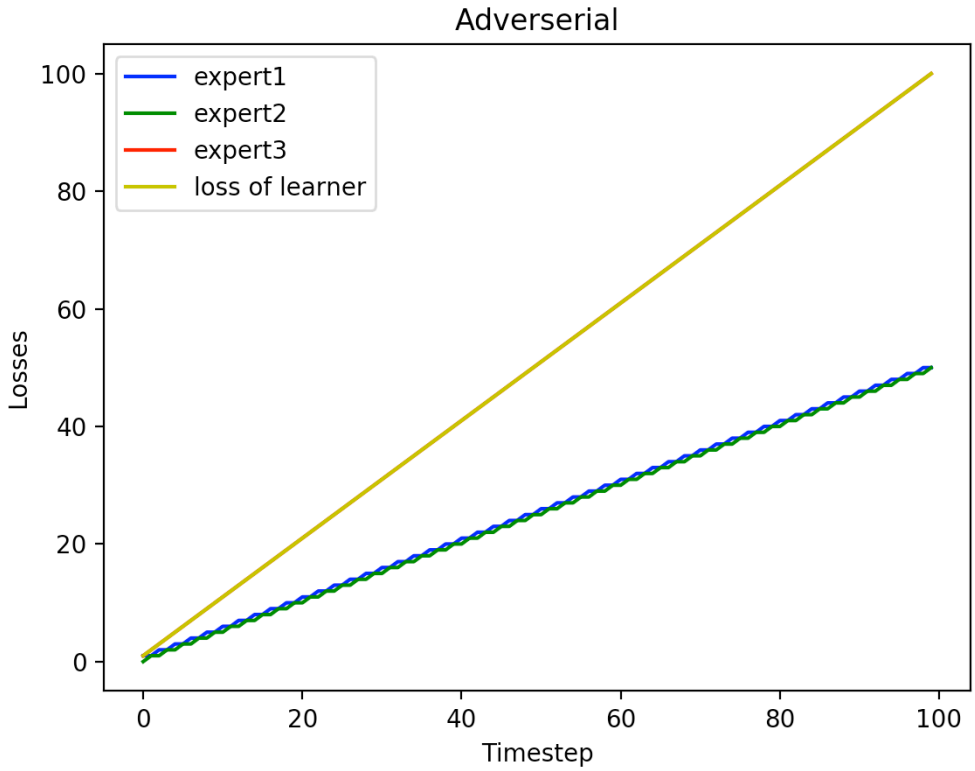


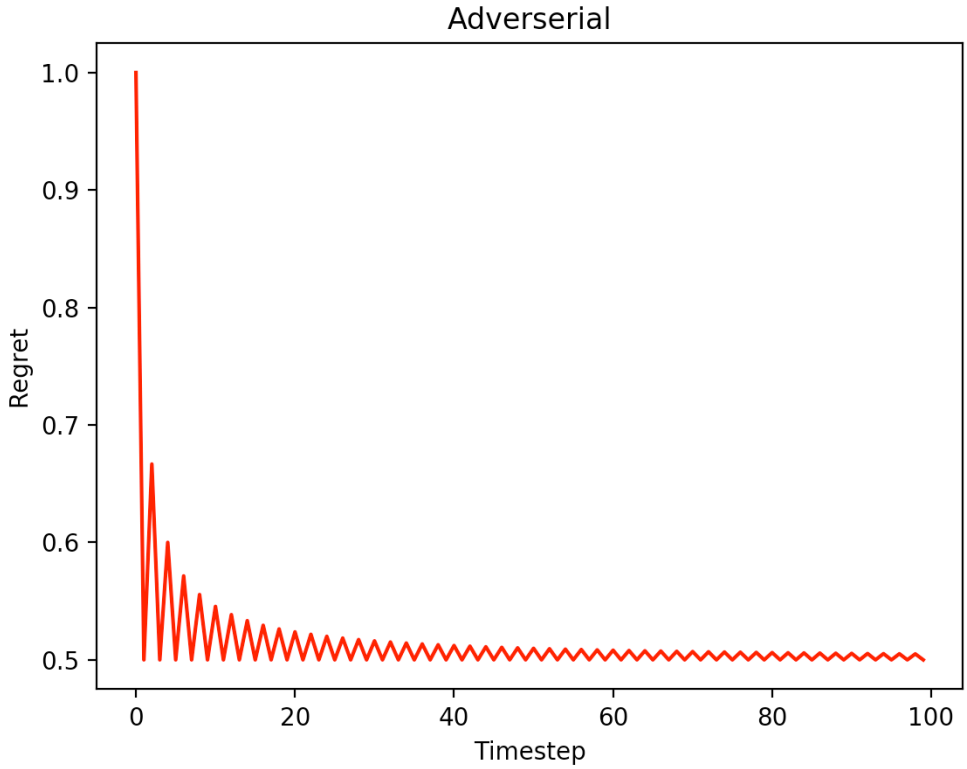
Now onto deterministic at $\eta = \frac{1}{10}$



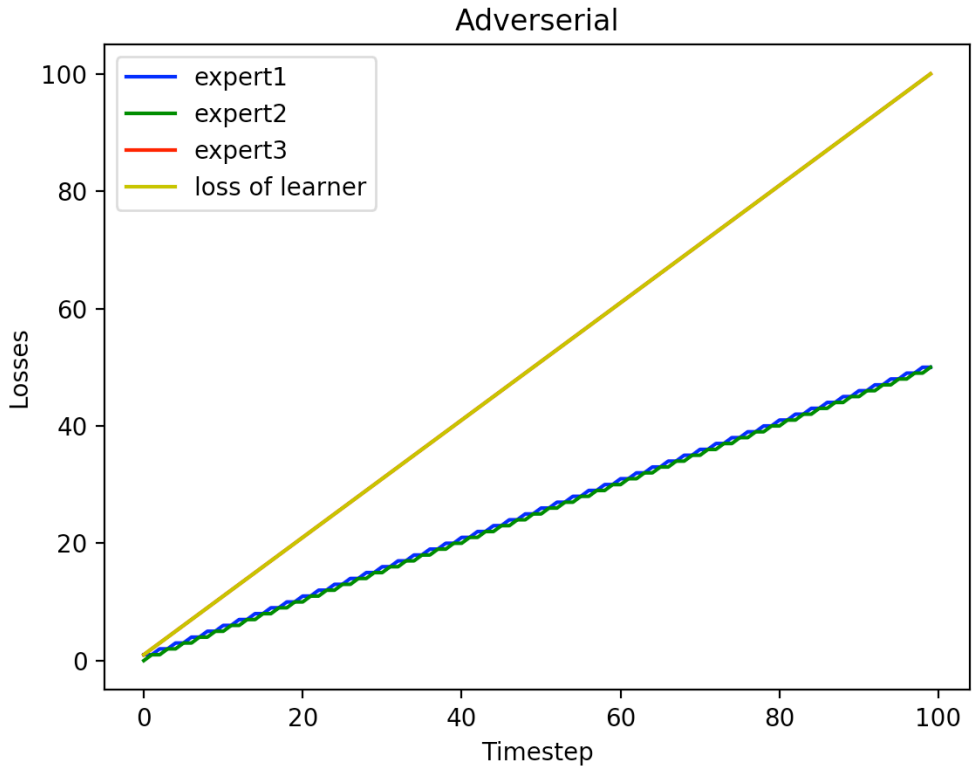


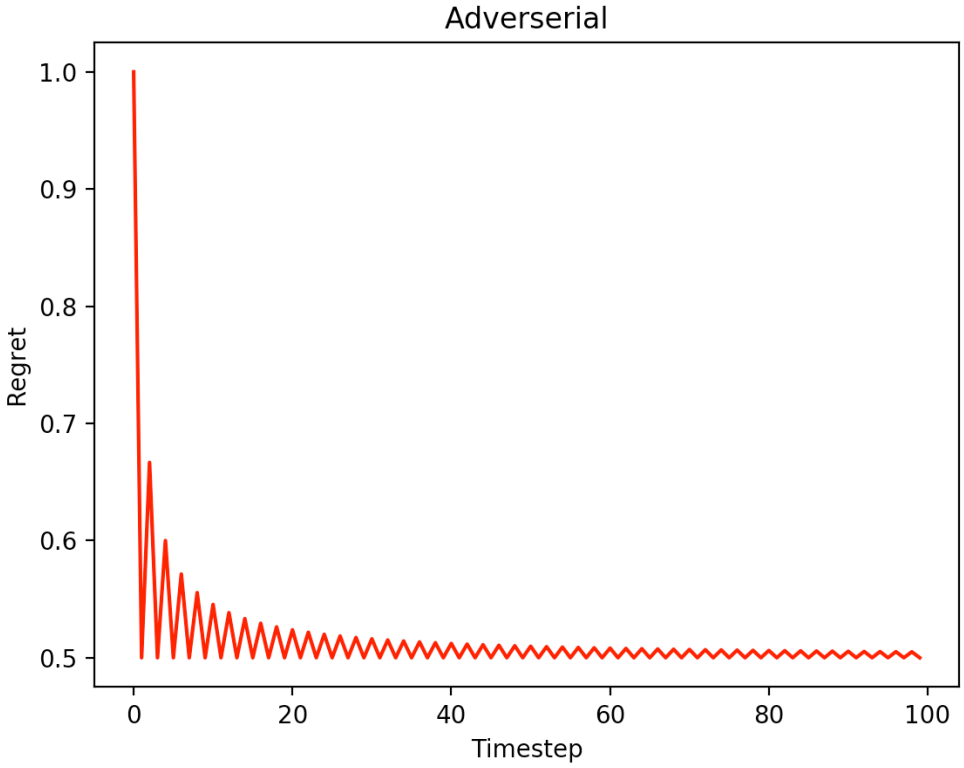
Now onto adversarial at $\eta = \frac{1}{2}$





Now onto adversarial at $\eta = \frac{1}{10}$

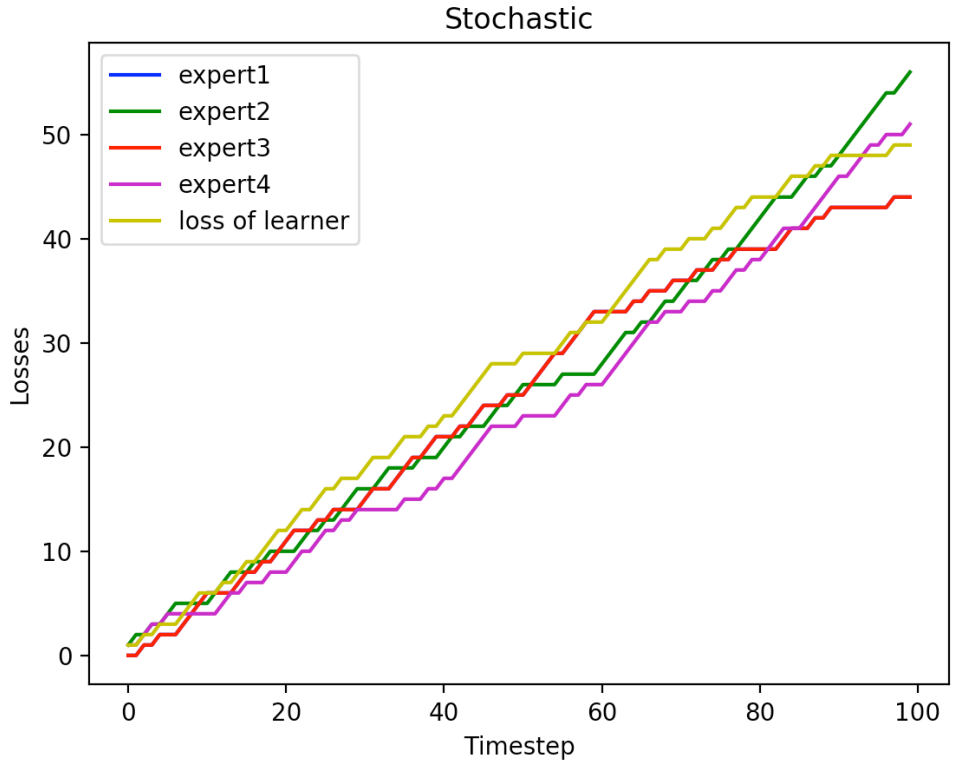


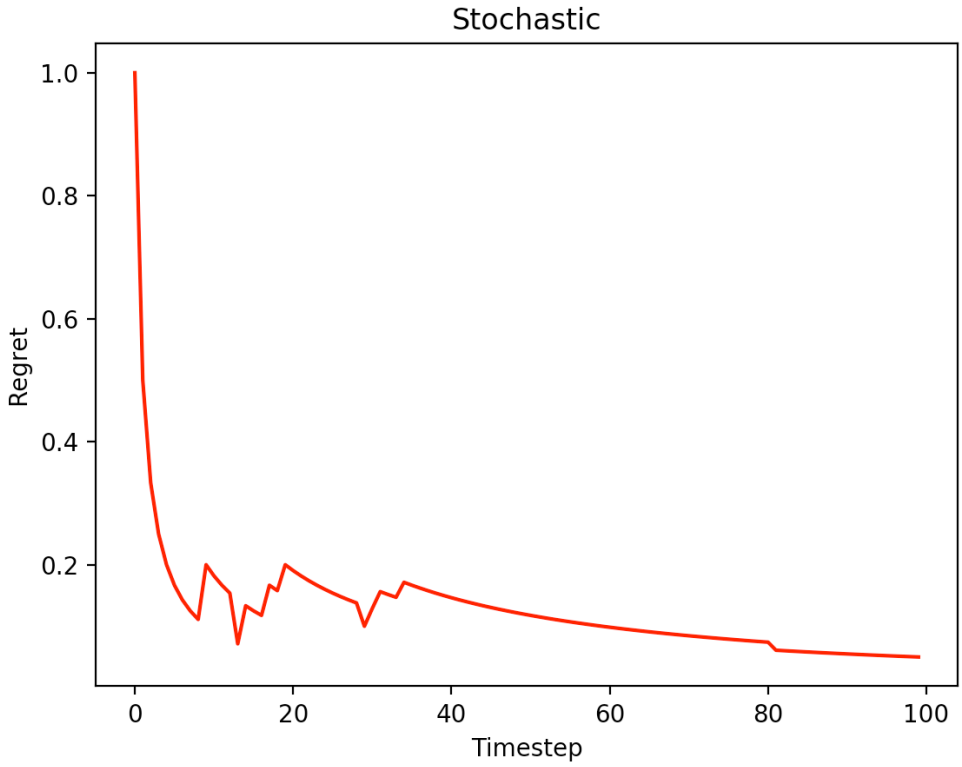


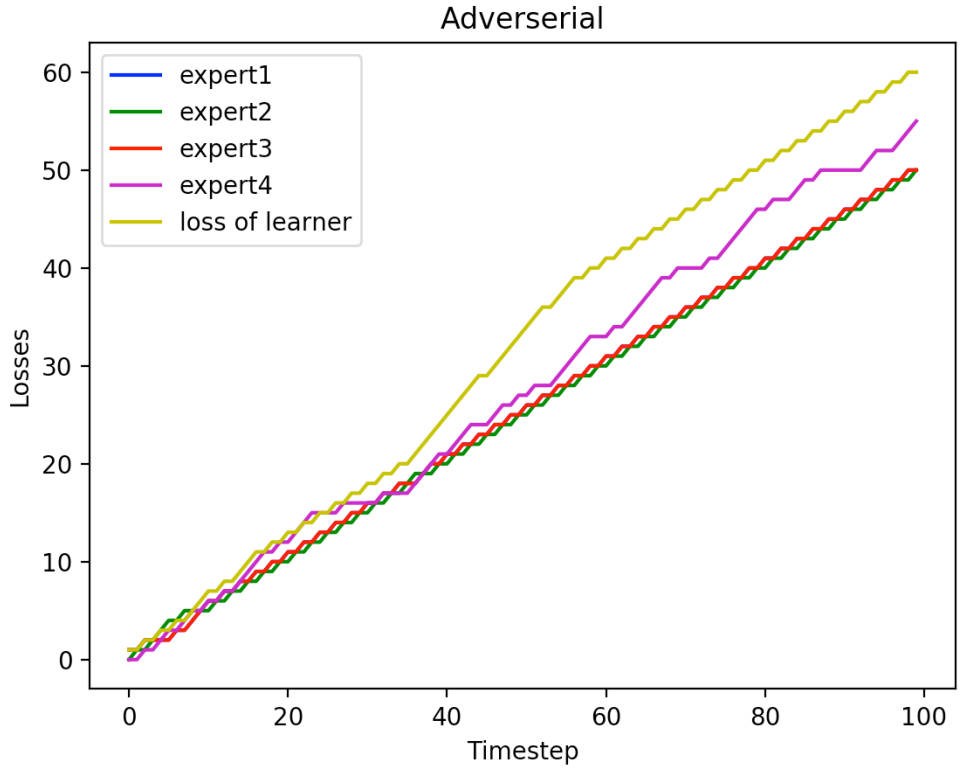
A few quick notes about these graphs:

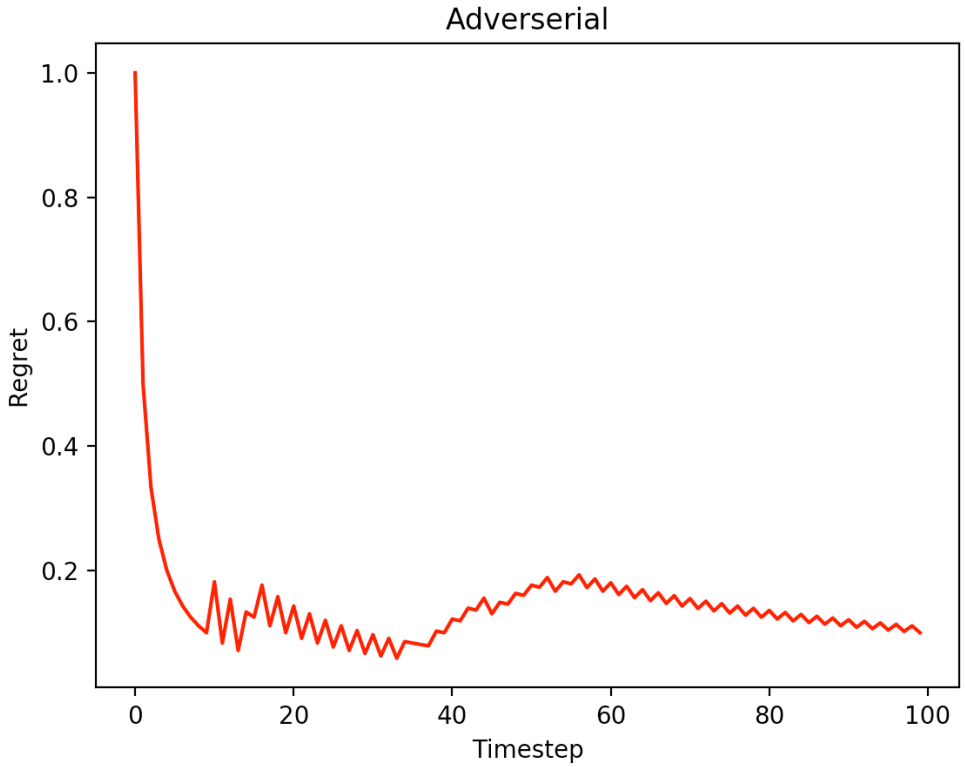
- The deterministic output is slightly skewed in favour of expert 1 as the output is goes like 0,0,1,1,1,1,0,0,1,1,1,1,0,0 etc, this explains why expert 1 has a lower loss than the others during the deterministic output nature.
- Now another thing to notice is that the loss gradient is steeper for higher η
- The regret also tends to asymptote towards η with each timestep

3.5 WMA With One additional expert who predicts that Tartans will win if it is sunny and lose if it rains.

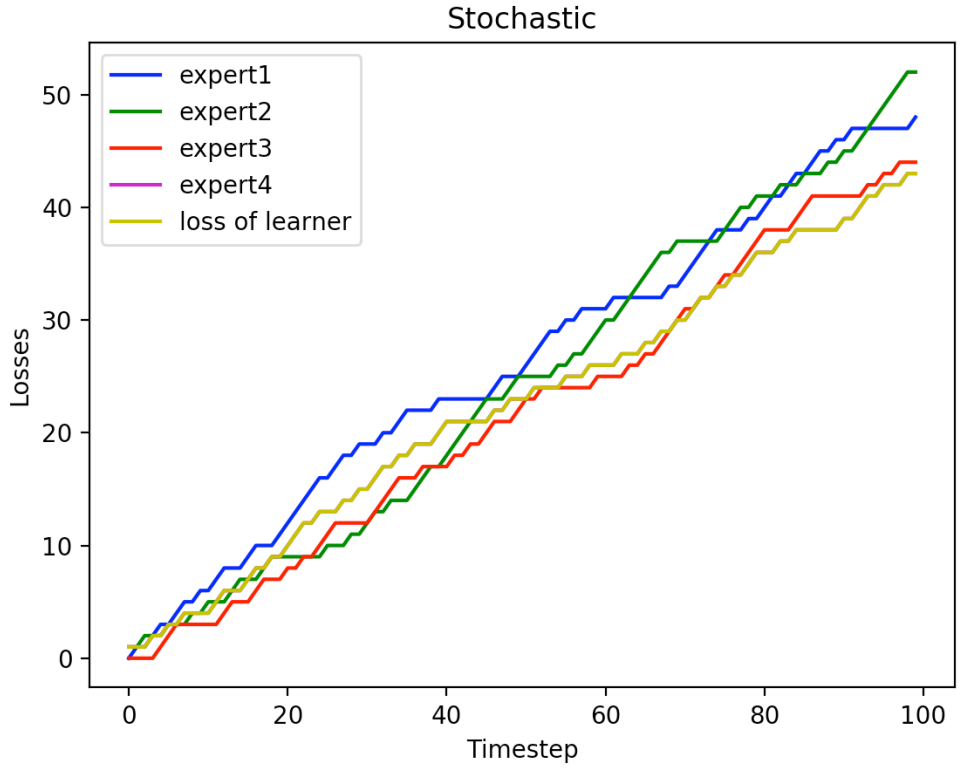


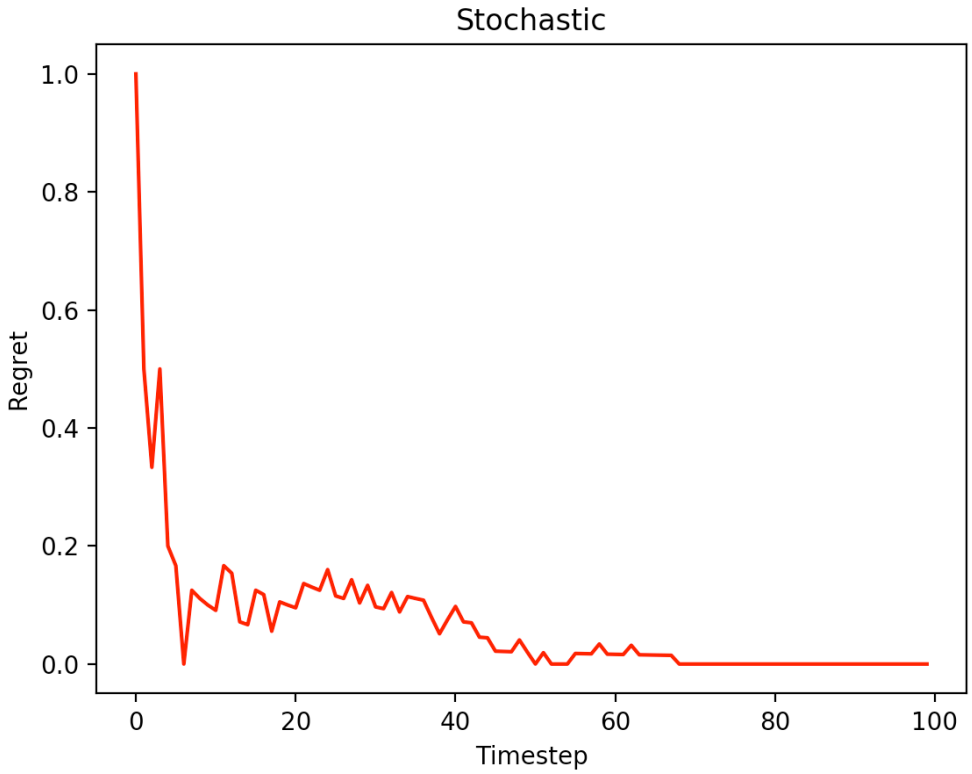


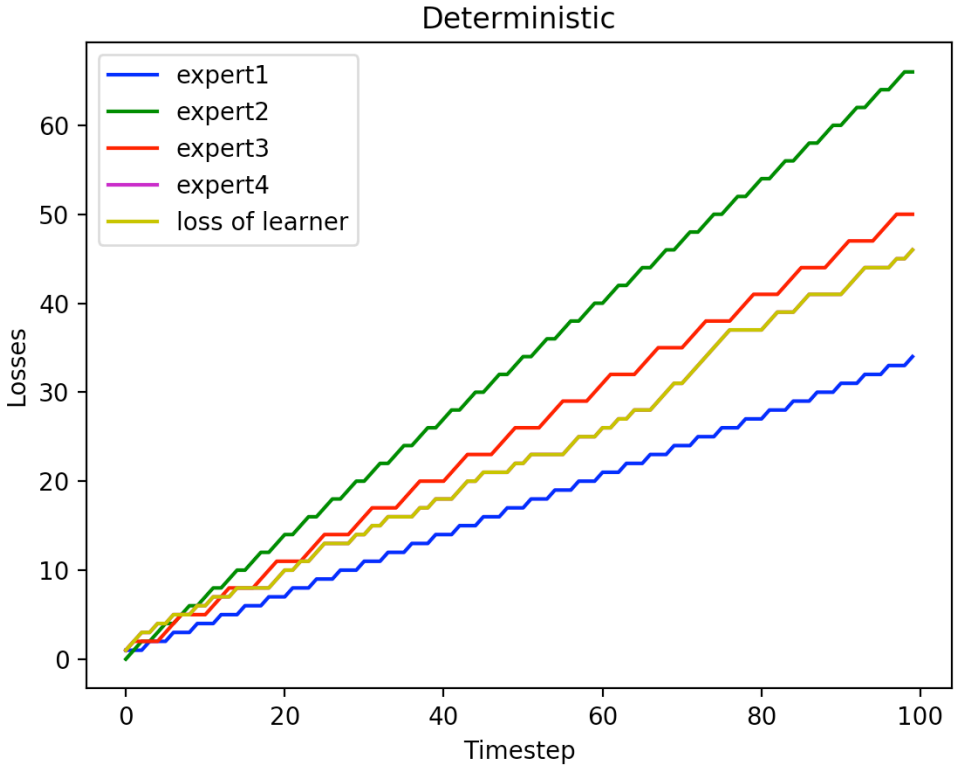

Adverserial
expert1
expert2
expert3
expert4
loss of learner
Losses
Timestep
0
10
20
30
40
50
60
0
20
40
60
80
100

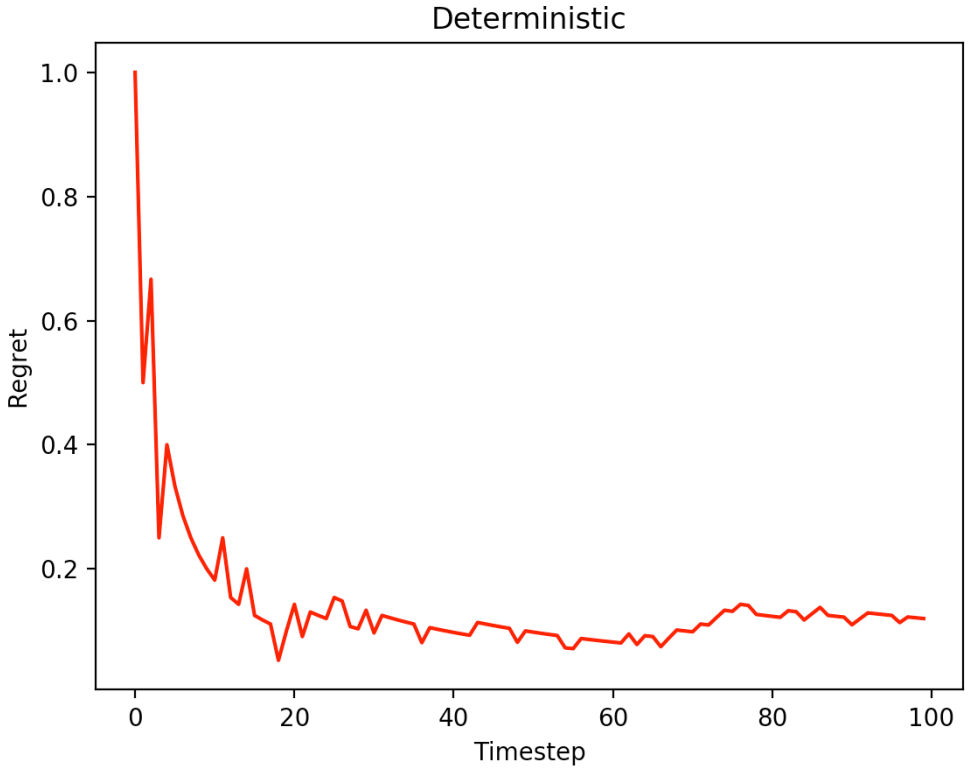

Adverserial
Regret
Timestep
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100

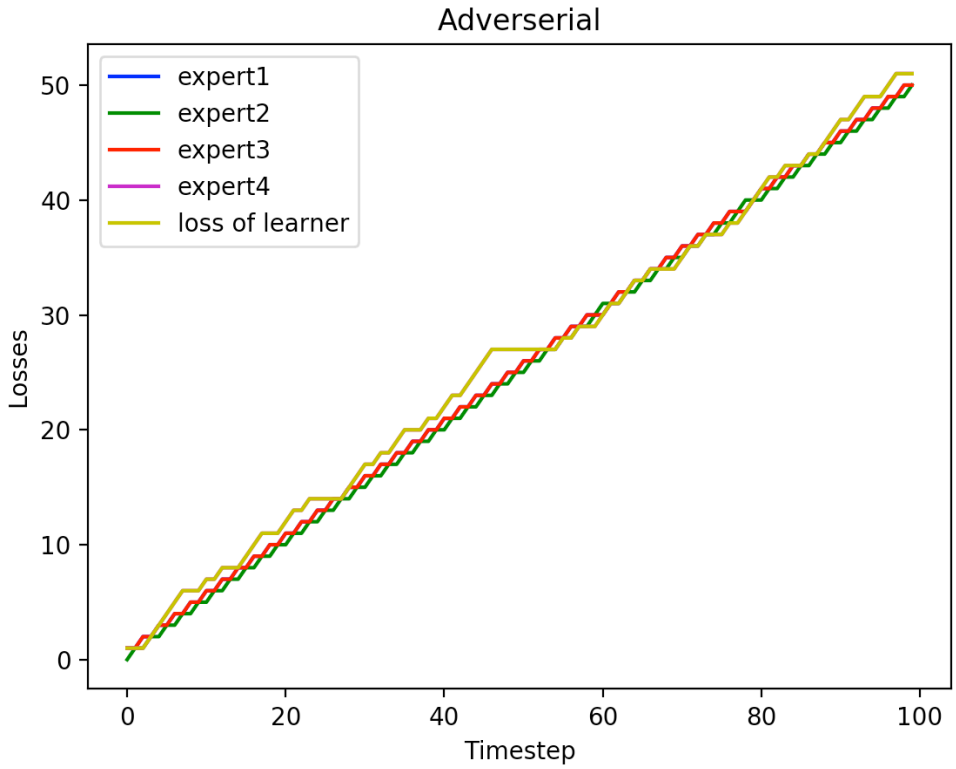
3.5 RWMA With One additional expert who predicts that Tartans will win if it is sunny and lose if it rains.

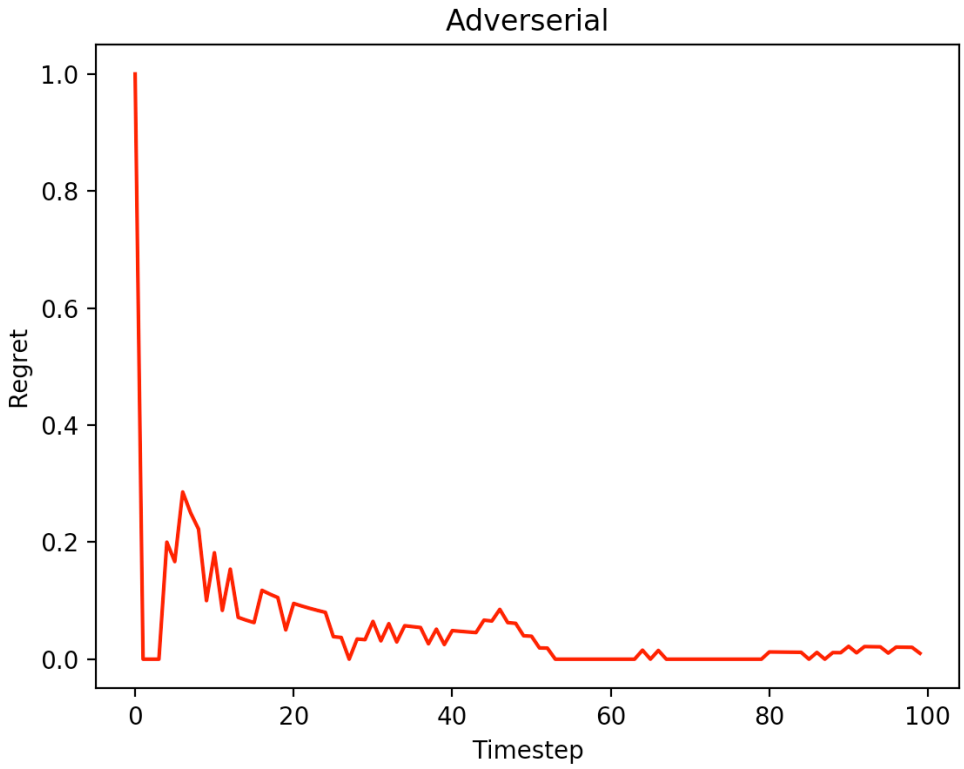





Deterministic
expert1
expert2
expert3
expert4
loss of learner
Losses
Timestep
0
10
20
30
40
50
60
0
20
40
60
80
100


Deterministic
Regret
Timestep
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100





Based on this output, several observations can be made. The first is that adding additional extraneous observations tightens the bound provided that the probability of classes per feature is homogenous (equal) across all features.