16831 Homework 1 - Theory

Jennifer Isaza

TOTAL POINTS

95.5 / 100

QUESTION 1

1 Introduction **0 / 0**

✓ **- 0 pts** **Correct**

QUESTION 2

Theory 45 pts

2.1 Regret **5 / 5**

✓ **- 0 pts** **Correct**

2.2 Constructing experts **5 / 5**

✓ **- 0 pts** **Correct**

2.3 Understanding penalty parameter **7.5 / 10**

✓ **- 2.5 pts** **Part 3: Question not answered for WMA**

2.4 Multi-class classification **15 / 15**

✓ **- 0 pts** **Correct**

2.5 Understanding adversarial environments **8 / 10**

✓ **- 2 pts** **Part 2: missing/incorrect mathematical reasoning**

Part 2 (-2 points): There is a reason why your first sentence leads to strictly better performance; you have not completed this point. You have not mathematically derived why a randomized algorithm is better when at least one expert is correct at least once.

QUESTION 3

Coding 55 pts

3.1 Programming nature **10 / 10**

✓ **- 0 pts** **Correct**

3.2 Implement WMA **15 / 15**

✓ **- 0 pts** **Correct**

3.3 Implement RWMA **15 / 15**

✓ **- 0 pts** **Correct**

3.4 More experts and observations/features **15 / 15**

✓ **- 0 pts** **Correct**

gradescope

HW1

Jennifer Isaza
Statistical Methods in Robotics (16-831)

September 20, 2018

Exercise 2.1. Regret: Can regret be negative?

Yes, regret can be negative. If the solution is unrealizable, the best expert could always be wrong, which means that the learner could have fewer mistakes than the best expert.

Exercise 2.2. Constructing Experts

A hypothesis class that will always have a correct mapping for a finite number of inputs to a finite number of outputs is possible. In order for the class to always produce true labels, there must be an expert that corresponds to each input for each output. In other words, there must be k^n experts for k=outputs and n=inputs.

Exercise 2.3. Understanding Penalty Parameter η

1.) Weighted Majority Algorithm:

$$R_W^* \leq (1+2\eta)m^* + \frac{2\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq 2m^* + (-2\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

Random Weighted Majority Algorithm:

$$E(R_W^*) \leq \eta m^* + \frac{\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq m^* + (-\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

1 Introduction 0 / 0

✓ - 0 pts Correct

gradescope

HW1

Jennifer Isaza
Statistical Methods in Robotics (16-831)

September 20, 2018

Exercise 2.1. Regret: Can regret be negative?

Yes, regret can be negative. If the solution is unrealizable, the best expert could always be wrong, which means that the learner could have fewer mistakes than the best expert.

Exercise 2.2. Constructing Experts

A hypothesis class that will always have a correct mapping for a finite number of inputs to a finite number of outputs is possible. In order for the class to always produce true labels, there must be an expert that corresponds to each input for each output. In other words, there must be k^n experts for k=outputs and n=inputs.

Exercise 2.3. Understanding Penalty Parameter η

1.) Weighted Majority Algorithm:

$$R_W^* \leq (1 + 2\eta)m^* + \frac{2\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq 2m^* + (-2\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

Random Weighted Majority Algorithm:

$$E(R_W^*) \leq \eta m^* + \frac{\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq m^* + (-\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

2.1 Regret 5 / 5

✓ - 0 pts Correct

gradescope

HW1

Jennifer Isaza
Statistical Methods in Robotics (16-831)

September 20, 2018

Exercise 2.1. Regret: Can regret be negative?

Yes, regret can be negative. If the solution is unrealizable, the best expert could always be wrong, which means that the learner could have fewer mistakes than the best expert.

Exercise 2.2. Constructing Experts

A hypothesis class that will always have a correct mapping for a finite number of inputs to a finite number of outputs is possible. In order for the class to always produce true labels, there must be an expert that corresponds to each input for each output. In other words, there must be k^n experts for k=outputs and n=inputs.

Exercise 2.3. Understanding Penalty Parameter η

1.) Weighted Majority Algorithm:

$$R_W^* \leq (1+2\eta)m^* + \frac{2\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq 2m^* + (-2\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

Random Weighted Majority Algorithm:

$$E(R_W^*) \leq \eta m^* + \frac{\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq m^* + (-\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

2.2 Constructing experts 5 / 5

✓ **- 0 pts** Correct

gradescope

HW1

Jennifer Isaza
Statistical Methods in Robotics (16-831)

September 20, 2018

Exercise 2.1. Regret: Can regret be negative?

Yes, regret can be negative. If the solution is unrealizable, the best expert could always be wrong, which means that the learner could have fewer mistakes than the best expert.

Exercise 2.2. Constructing Experts

A hypothesis class that will always have a correct mapping for a finite number of inputs to a finite number of outputs is possible. In order for the class to always produce true labels, there must be an expert that corresponds to each input for each output. In other words, there must be k^n experts for k=outputs and n=inputs.

Exercise 2.3. Understanding Penalty Parameter η

1.) Weighted Majority Algorithm:

$$R_W^* \leq (1+2\eta)m^* + \frac{2\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq 2m^* + (-2\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

Random Weighted Majority Algorithm:

$$E(R_W^*) \leq \eta m^* + \frac{\ln N}{\eta}$$

$$\frac{\partial R_W^*}{\partial \eta} : 0 \leq m^* + (-\ln N)\eta^{-2}$$

$$\eta = \sqrt{\frac{\ln N}{m^*}}$$

2.)

We should choose a larger value of η. If m^* is very small compared to N, the term "$\frac{\ln N}{\eta}$" will dominate the equation. If we want to minimize regret, η should be larger in order to decrease the dominant part of the equation. Since η has the same relation for both WMA and RWMA, η should be a larger value for both equations.

3.)

(a) m* = O(T)
Yes, we can choose an η to make the algorithm no-regret. Choosing ($\eta = \frac{1}{\sqrt{T}}$) will make both ηm* and $\frac{\ln N}{\eta}$ sublinear

(b) m* = O(T^x)
Yes, we can choose an η to make the algorithm no-regret. Choosing (η = T) will make both ηm* and $\frac{\ln N}{\eta}$ sublinear.

Exercise 2.4. Multi-Class Classification

1.) For the fully observable scenario with k classes, the version space will decrease by at least (1 - $\frac{1}{k}$) for every mistake. Similarly, for the Halving algorithm, the version space will decrease by the same relationship, when setting k = 2. The smaller k is, the slower the version space will decrease, which means more opportunities for mistakes. Therefore, the upper bound on mistakes will also be when $k = 2$ for the multi-class situation.

Bound Derivation:

$$|V^{t+1}| \leq (\frac{1}{k})|V^t|$$
$$|V^t| \leq (\frac{1}{k})^M E$$
$$1 \leq |V^t| \leq k^{-M} E$$
$$M \leq log_k E$$
$$k = 2 : M \leq log_2 E$$

2.3 Understanding penalty parameter **7.5 / 10**

✓ **- 2.5 pts Part 3: Question not answered for WMA**

gradescope

2.)

We should choose a larger value of η. If m^* is very small compared to N, the term "$\frac{\ln N}{\eta}$" will dominate the equation. If we want to minimize regret, η should be larger in order to decrease the dominant part of the equation. Since η has the same relation for both WMA and RWMA, η should be a larger value for both equations.

3.)

(a) m* = O(T)
Yes, we can choose an η to make the algorithm no-regret. Choosing ($\eta = \frac{1}{\sqrt{T}}$) will make both ηm* and $\frac{\ln N}{\eta}$ sublinear

(b) m* = O(T^x)
Yes, we can choose an η to make the algorithm no-regret. Choosing (η = T) will make both ηm* and $\frac{\ln N}{\eta}$ sublinear.

Exercise 2.4. Multi-Class Classification

1.) For the fully observable scenario with k classes, the version space will decrease by at least (1 - $\frac{1}{k}$) for every mistake. Similarly, for the Halving algorithm, the version space will decrease by the same relationship, when setting k = 2. The smaller k is, the slower the version space will decrease, which means more opportunities for mistakes. Therefore, the upper bound on mistakes will also be when $k = 2$ for the multi-class situation.

Bound Derivation:

$$|V^{t+1}| \leq (\frac{1}{k})|V^t|$$
$$|V^t| \leq (\frac{1}{k})^M E$$
$$1 \leq |V^t| \leq k^{-M} E$$
$$M \leq log_k E$$
$$k = 2 : M \leq log_2 E$$

2.) For the partially observable scenario with k classes, the version space will decrease by at least $(\frac{1}{k})$ for every mistake, since the learner is only told if one class is correct or not, but has no information about the other classes.

$$
\begin{aligned}
& V^{(1)} = H \\
& fort = 1, ..., Tdo \\
& \quad Receive(x^(t)) \\
& \quad \hat{y} = MajorityConsensus(V^t, x^{(t)}) \\
& \quad Receive(y^t) \\
& \quad if(y^t ! = \hat{y}) \\
& \quad\quad V^{(t+1)} = h \in V^{(t)} : h(x^{(t)})! = \hat{y}
\end{aligned}
$$

Bound Derivation:

$$
\begin{aligned}
|V^{t+1}| &\leq (\frac{k-1}{k})|V^t| \\
|V^t| &\leq (\frac{k-1}{k})^M E \\
1 \leq |V^t| &\leq (\frac{k}{k-1})^{-M} E \\
M &\leq log_{\frac{k}{k-1}} E
\end{aligned}
$$

Exercise 2.5. Understanding Adversarial Environments

1.) For the deterministic Weighted Majority Algorithm, the loss is maximized when the adversary chooses the opposite of the deterministic prediction. Since the adversary has access to the predictions and the weights of each expert, the learner's choice can be determined unless there is some form of randomized input.

2.) The adversarial can always calculate the opposite of the weighted majority algorithm, so the randomization factor will keep the adversarial from being able to calculate the opposite of the random weighted majority algorithm. This inability to predict the output of the expert predictions will lead to smaller losses when compared to the losses from the weighted majority algorithm experts.

Exercise 3.3. Implement the Weighted Majority Algorithm

Exercise 3.4. Implement the Random Weighted Majority Algorithm

2.4 Multi-class classification 15 / 15

✓ **- 0 pts** Correct

gradescope

2.) For the partially observable scenario with k classes, the version space will decrease by at least $(\frac{1}{k})$ for every mistake, since the learner is only told if one class is correct or not, but has no information about the other classes.

$$
\begin{aligned}
& V^{(1)} = H \\
& fort = 1, ..., Tdo \\
& \quad Receive(x^(t)) \\
& \quad \hat{y} = MajorityConsensus(V^t, x^{(t)}) \\
& \quad Receive(y^t) \\
& \quad if(y^t! = \hat{y}) \\
& \qquad V^{(t+1)} = h \in V^{(t)} : h(x^{(t)})! = \hat{y}
\end{aligned}
$$

Bound Derivation:

$$
\begin{aligned}
|V^{t+1}| &\leq (\frac{k-1}{k})|V^t| \\
|V^t| &\leq (\frac{k-1}{k})^M E \\
1 \leq |V^t| &\leq (\frac{k}{k-1})^{-M} E \\
M &\leq log_{\frac{k}{k-1}} E
\end{aligned}
$$

Exercise 2.5. Understanding Adversarial Environments

1.) For the deterministic Weighted Majority Algorithm, the loss is maximized when the adversary chooses the opposite of the deterministic prediction. Since the adversary has access to the predictions and the weights of each expert, the learner's choice can be determined unless there is some form of randomized input.

2.) The adversarial can always calculate the opposite of the weighted majority algorithm, so the randomization factor will keep the adversarial from being able to calculate the opposite of the random weighted majority algorithm. This inability to predict the output of the expert predictions will lead to smaller losses when compared to the losses from the weighted majority algorithm experts.

Exercise 3.3. Implement the Weighted Majority Algorithm

Exercise 3.4. Implement the Random Weighted Majority Algorithm

2.5 Understanding adversarial environments 8 / 10

✓ **- 2 pts Part 2: missing/incorrect mathematical reasoning**

Part 2 (-2 points): There is a reason why your first sentence leads to strictly better performance; you have not completed this point. You have not mathematically derived why a randomized algorithm is better when at least one expert is correct at least once.

gradescope

2.) For the partially observable scenario with k classes, the version space will decrease by at least $(\frac{1}{k})$ for every mistake, since the learner is only told if one class is correct or not, but has no information about the other classes.

$$
\begin{aligned}
& V^{(1)} = H \\
& for t = 1, ..., T do \\
& \quad Receive(x^(t)) \\
& \quad \hat{y} = MajorityConsensus(V^t, x^{(t)}) \\
& \quad Receive(y^t) \\
& \quad if(y^t ! = \hat{y}) \\
& \quad\quad V^{(t+1)} = h \in V^{(t)} : h(x^{(t)})! = \hat{y}
\end{aligned}
$$

Bound Derivation:

$$
\begin{aligned}
|V^{t+1}| &\leq (\frac{k-1}{k})|V^t| \\
|V^t| &\leq (\frac{k-1}{k})^M E \\
1 \leq |V^t| &\leq (\frac{k}{k-1})^{-M} E \\
M &\leq log_{\frac{k}{k-1}} E
\end{aligned}
$$

Exercise 2.5. Understanding Adversarial Environments

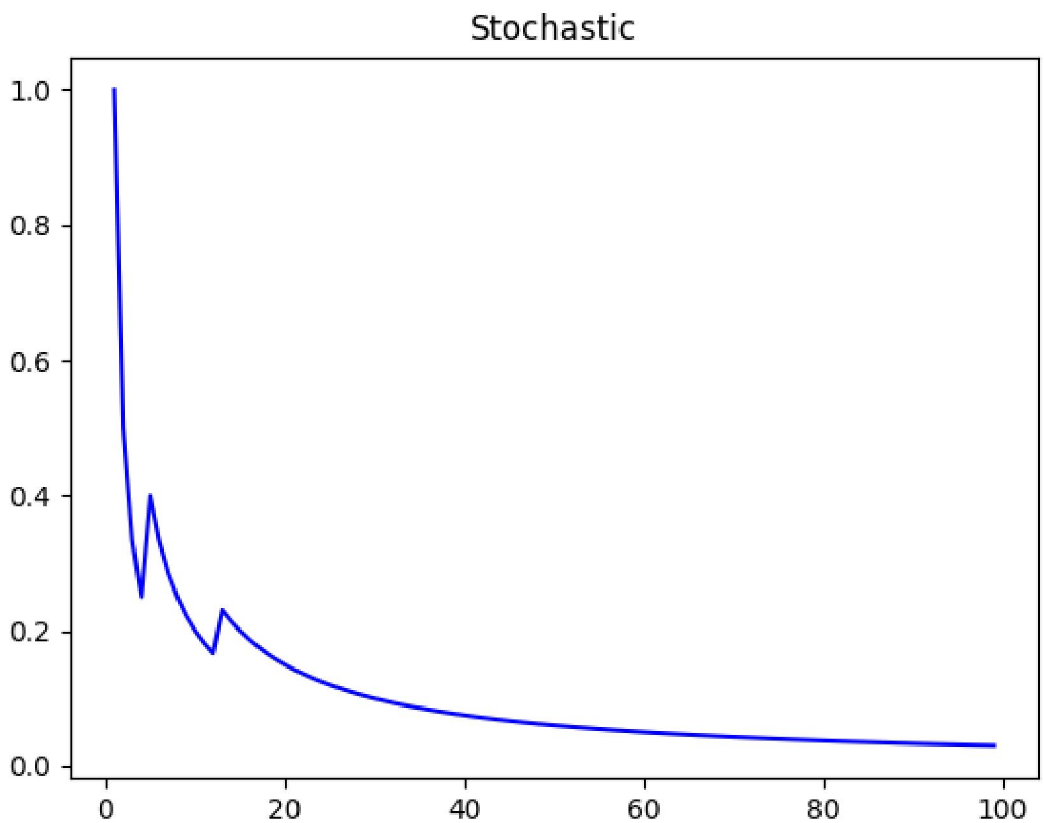
1.) For the deterministic Weighted Majority Algorithm, the loss is maximized when the adversary chooses the opposite of the deterministic prediction. Since the adversary has access to the predictions and the weights of each expert, the learner's choice can be determined unless there is some form of randomized input.

2.) The adversarial can always calculate the opposite of the weighted majority algorithm, so the randomization factor will keep the adversarial from being able to calculate the opposite of the random weighted majority algorithm. This inability to predict the output of the expert predictions will lead to smaller losses when compared to the losses from the weighted majority algorithm experts.

Exercise 3.3. Implement the Weighted Majority Algorithm

Exercise 3.4. Implement the Random Weighted Majority Algorithm

Figure 1: Weighted Majority Stochastic Graph



3.1 Programming nature 10 / 10

✓ **- 0 pts** Correct

gradescope

Figure 1: Weighted Majority Stochastic Graph

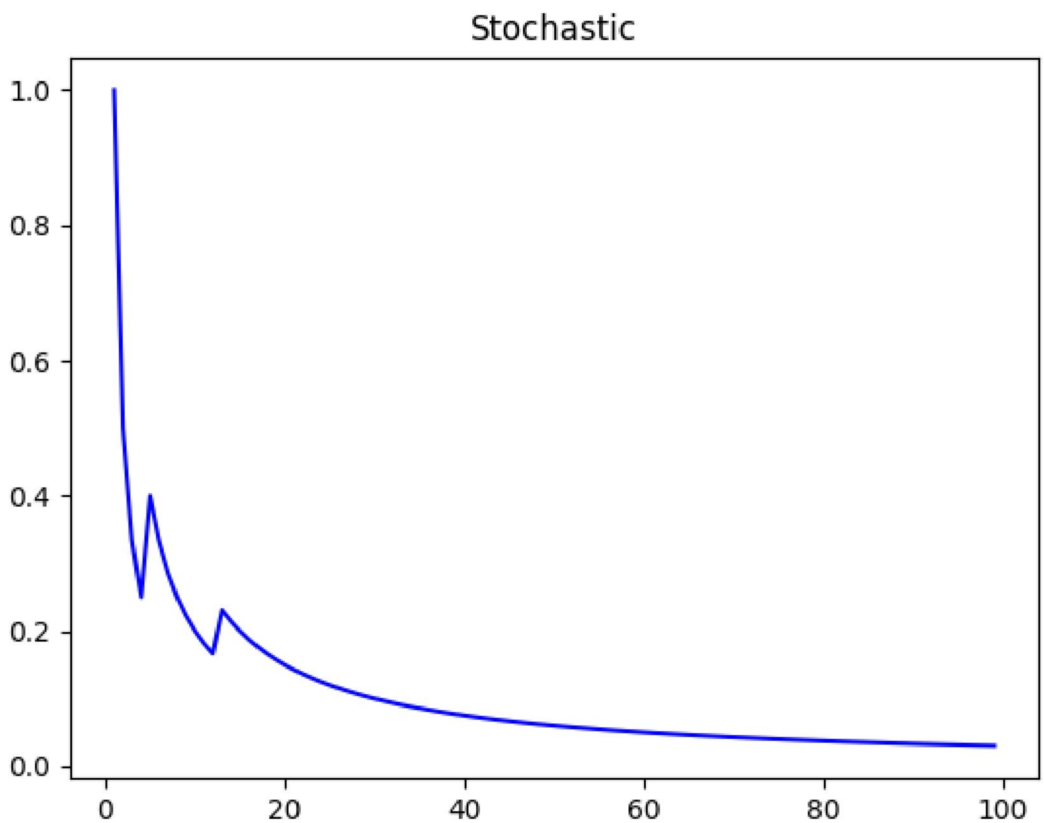


Figure 2: Weighted Majority Stochastic Expert Los Graph

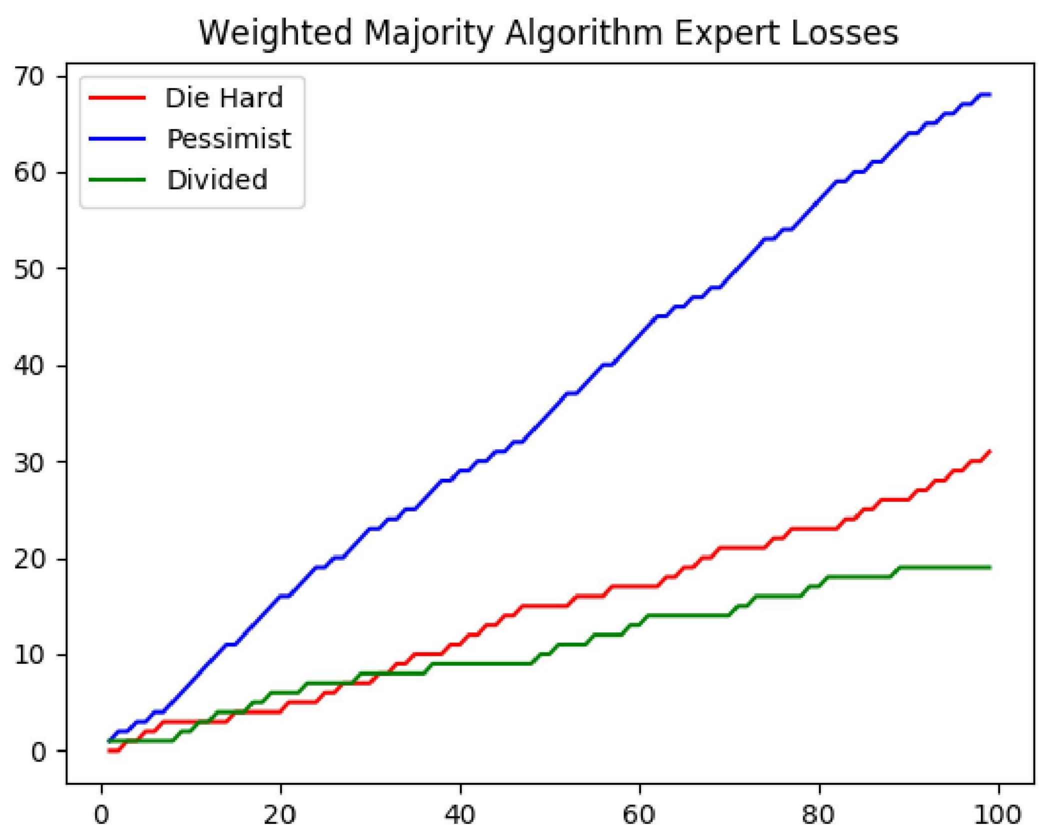


Figure 3: Weighted Majority Deterministic Graph

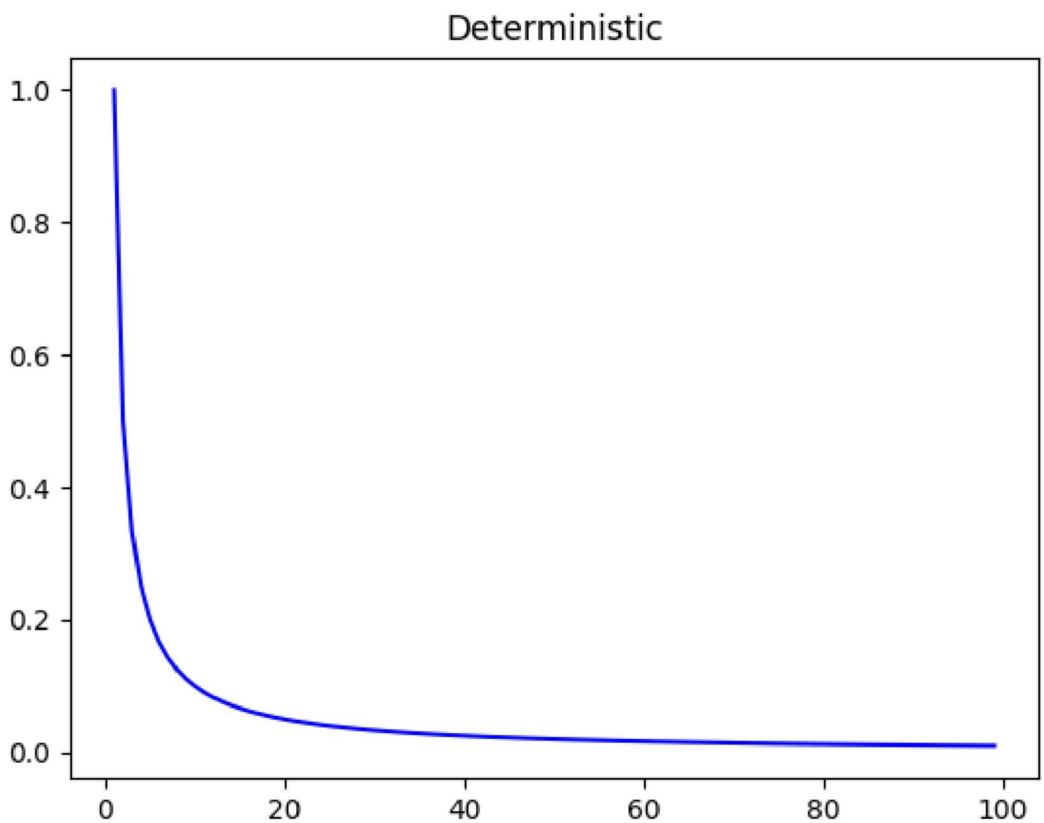


Figure 4: Weighted Majority Deterministic Expert Los Graph

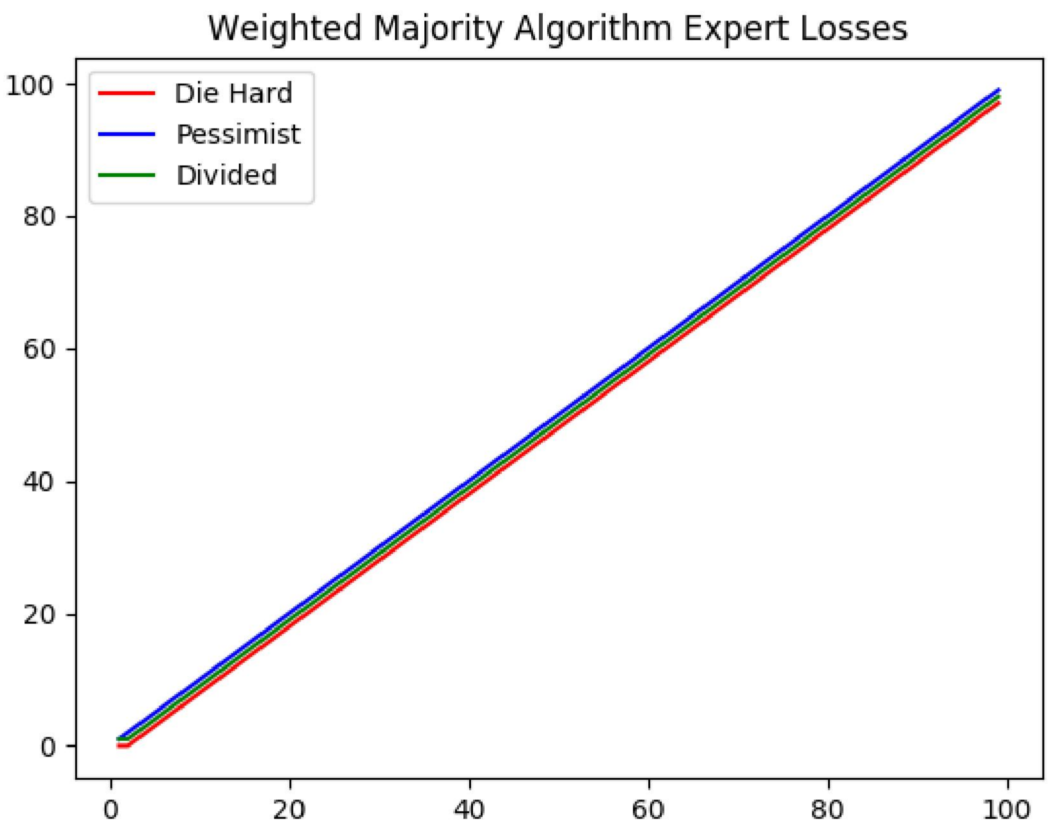


Figure 5: Weighted Majority Adversarial Graph

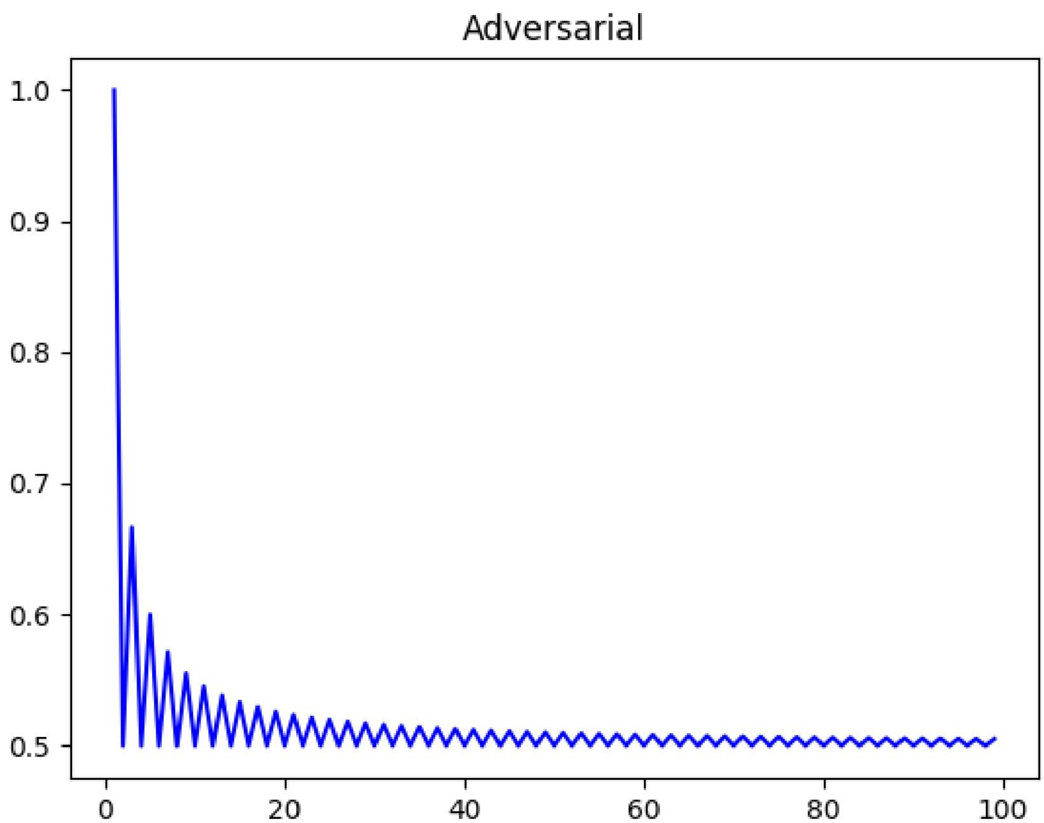
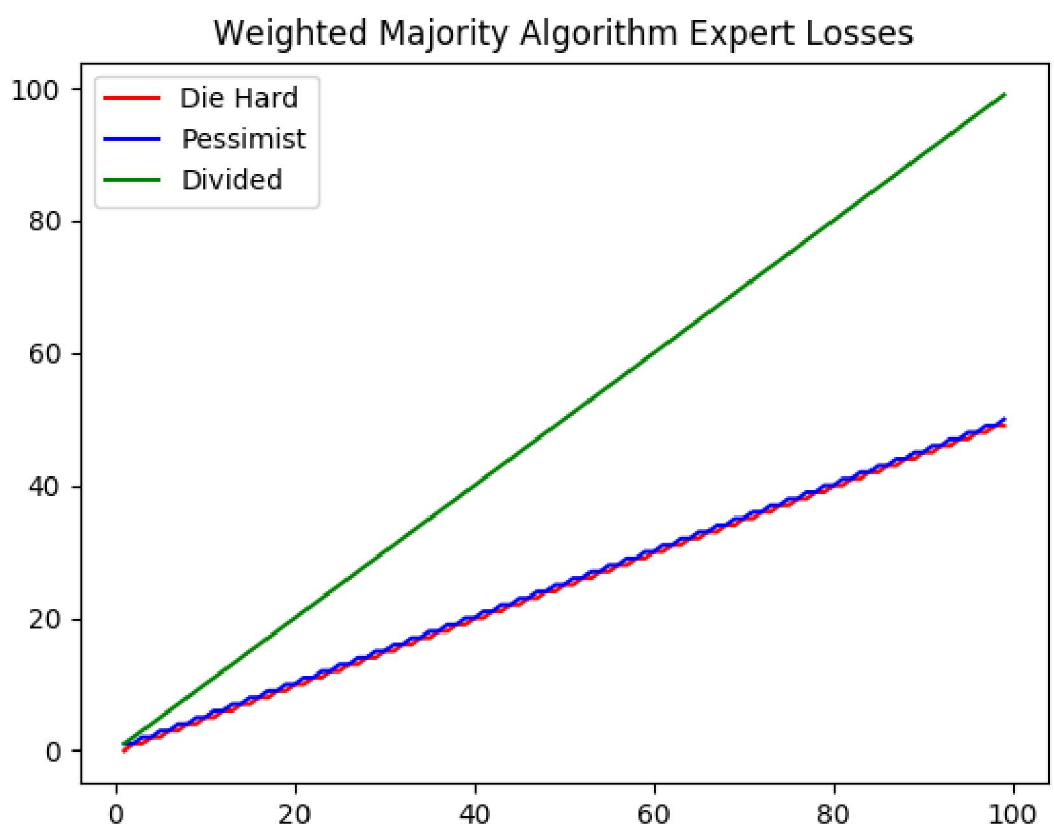


Figure 6: Weighted Majority Adversarial Expert Los Graph



3.2 Implement WMA 15 / 15

✓ **- 0 pts** Correct

gradescope

Figure 7: Random Weighted Majority Stochastic Graph

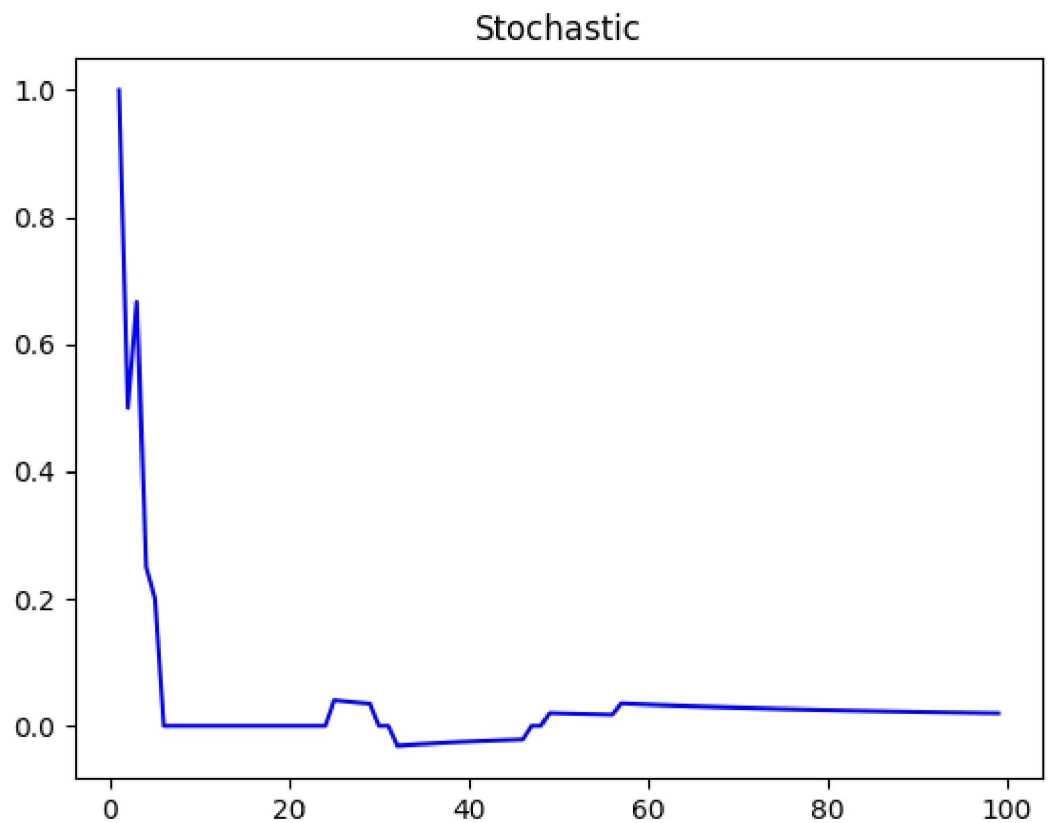


Figure 8: Random Weighted Majority Stochastic Expert Loss Graph

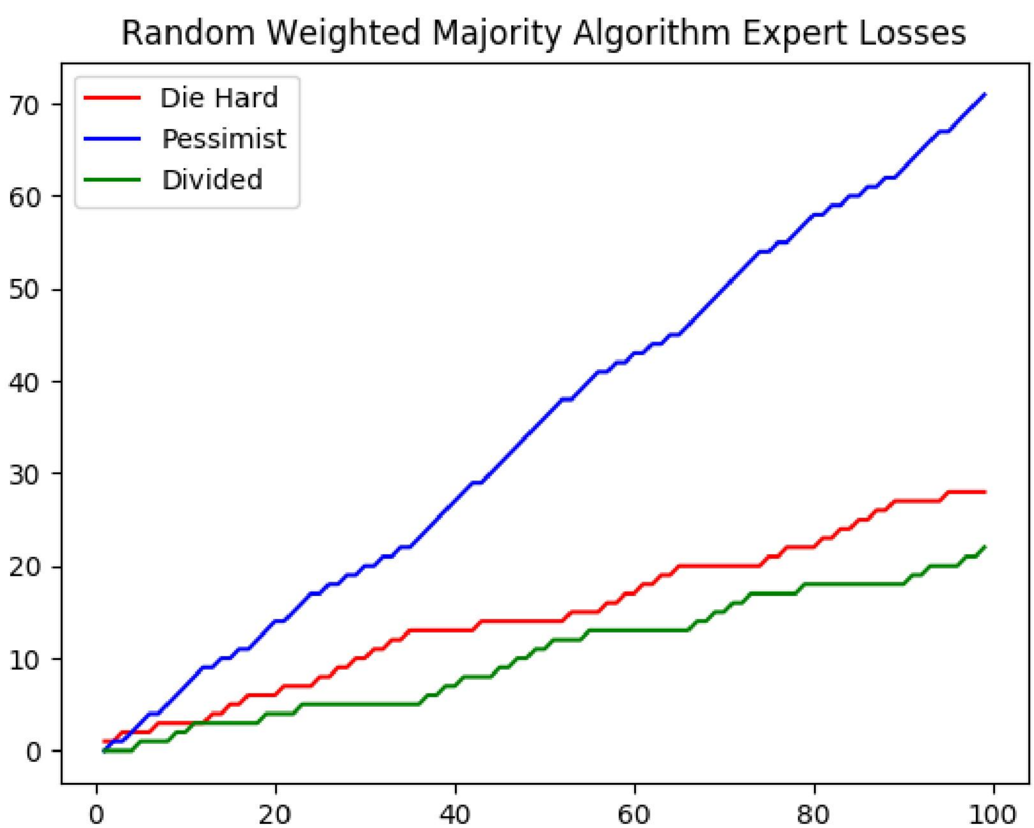


Figure 9: Random Weighted Majority Deterministic Graph

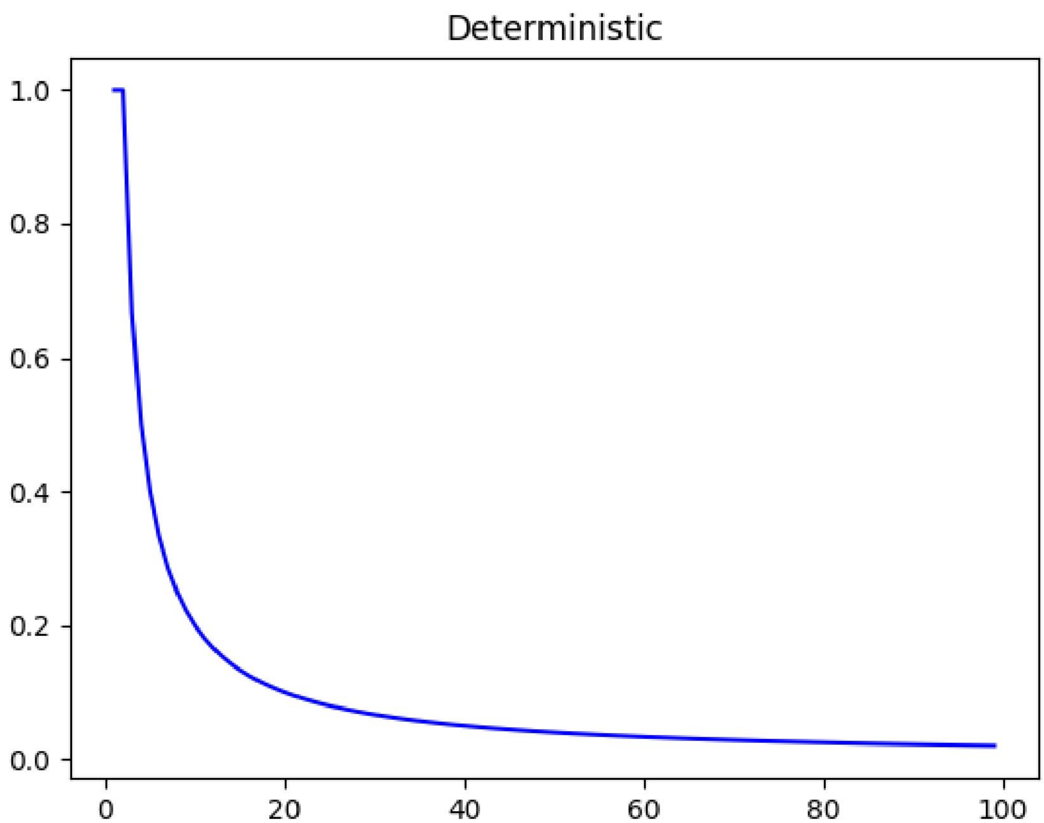


Figure 10: Random Weighted Majority Deterministic Expert Loss Graph

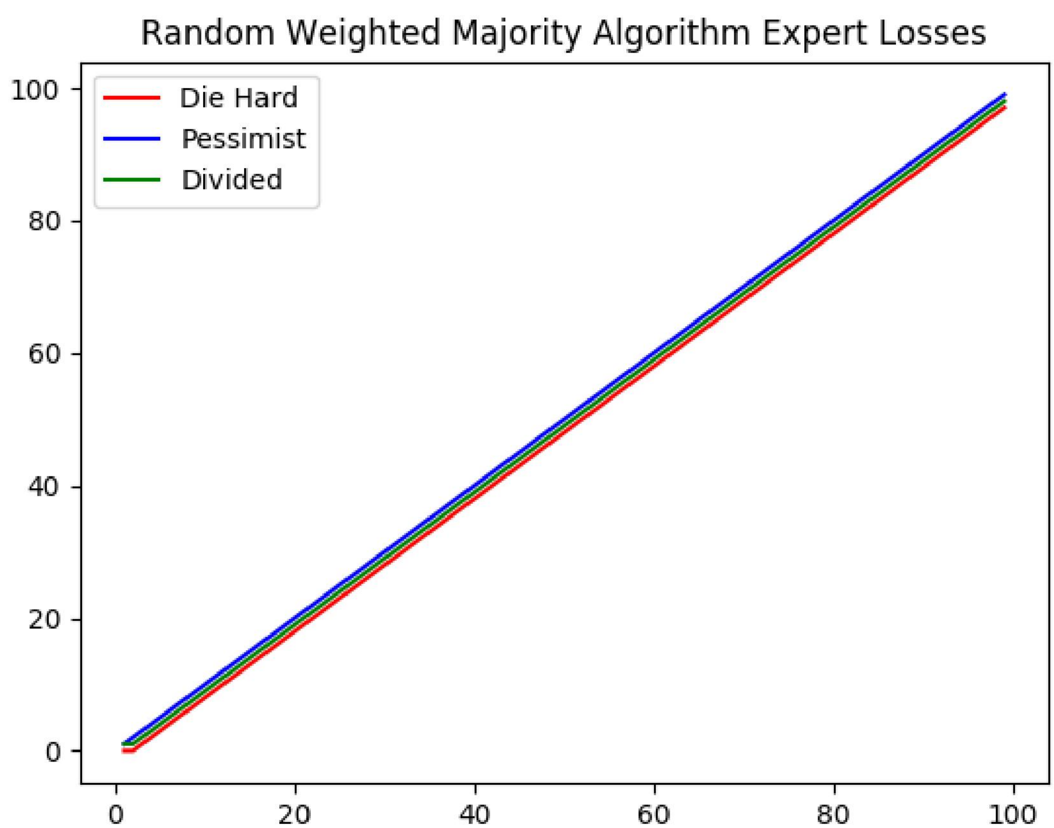


Figure 11: Random Weighted Majority Adversarial Graph

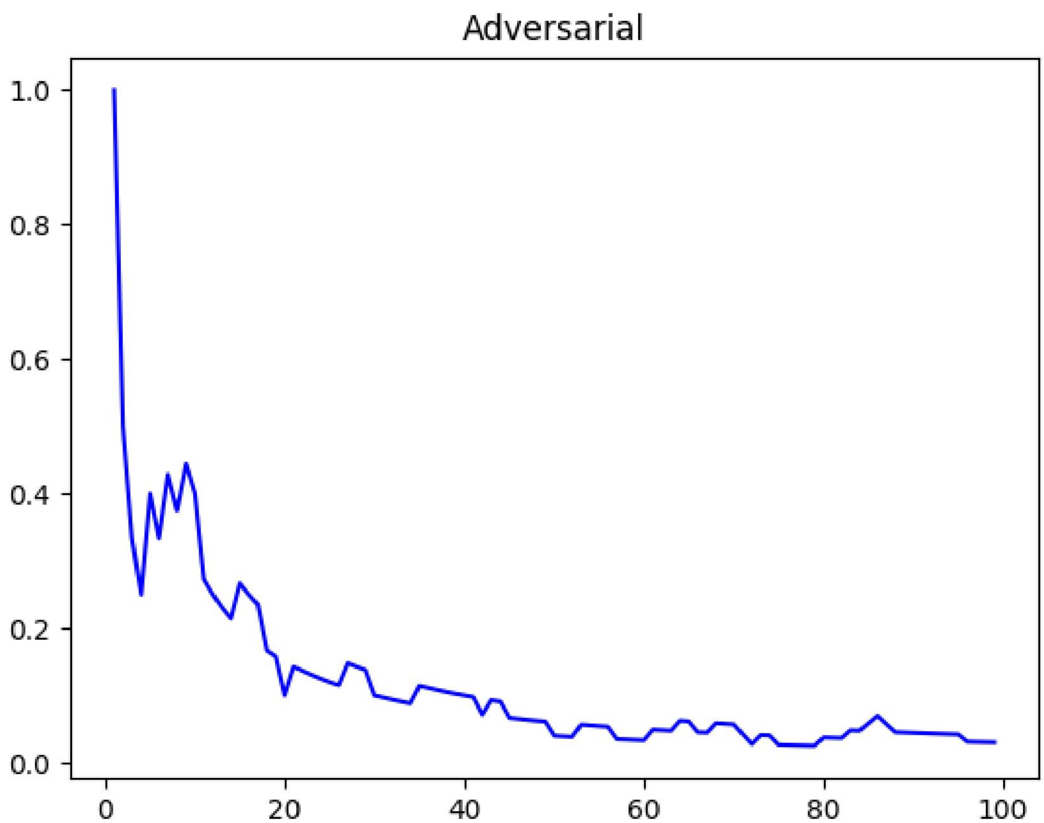
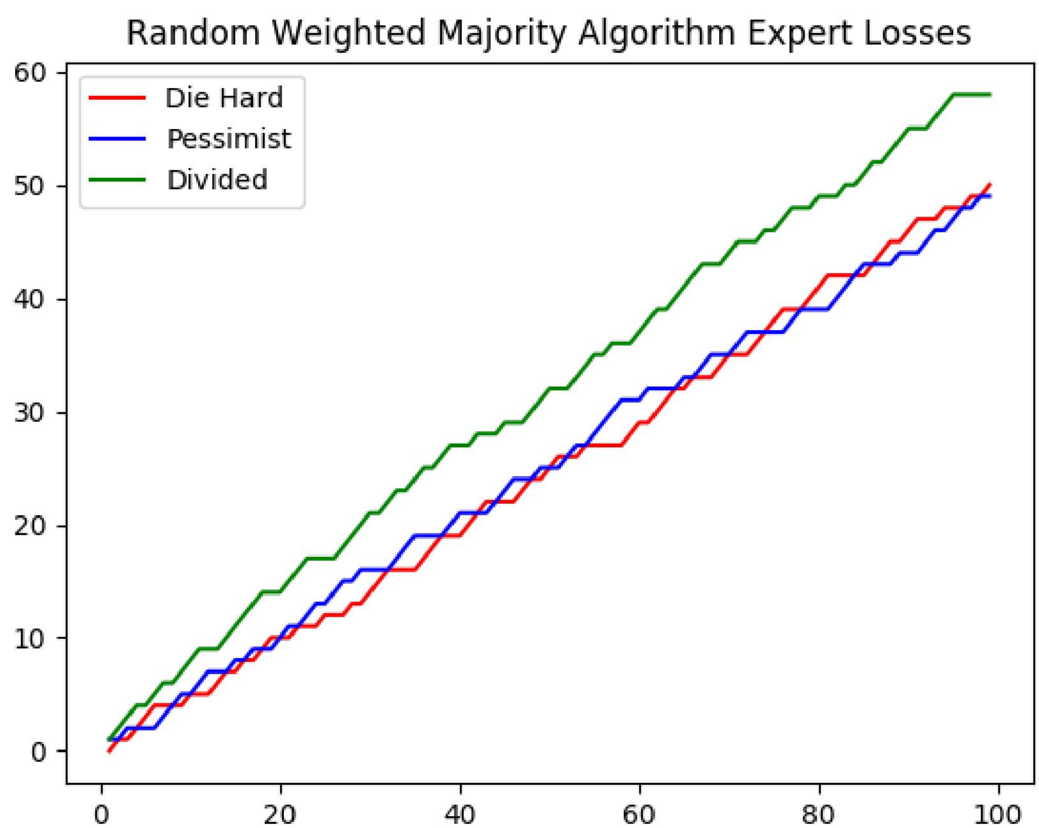


Figure 12: Random Weighted Majority Adversarial Expert Loss Graph



3.3 Implement RWMA 15 / 15

✓ **- 0 pts** Correct

gradescope

Figure 13: Weighted Majority Stochastic Graph using Extra Observations

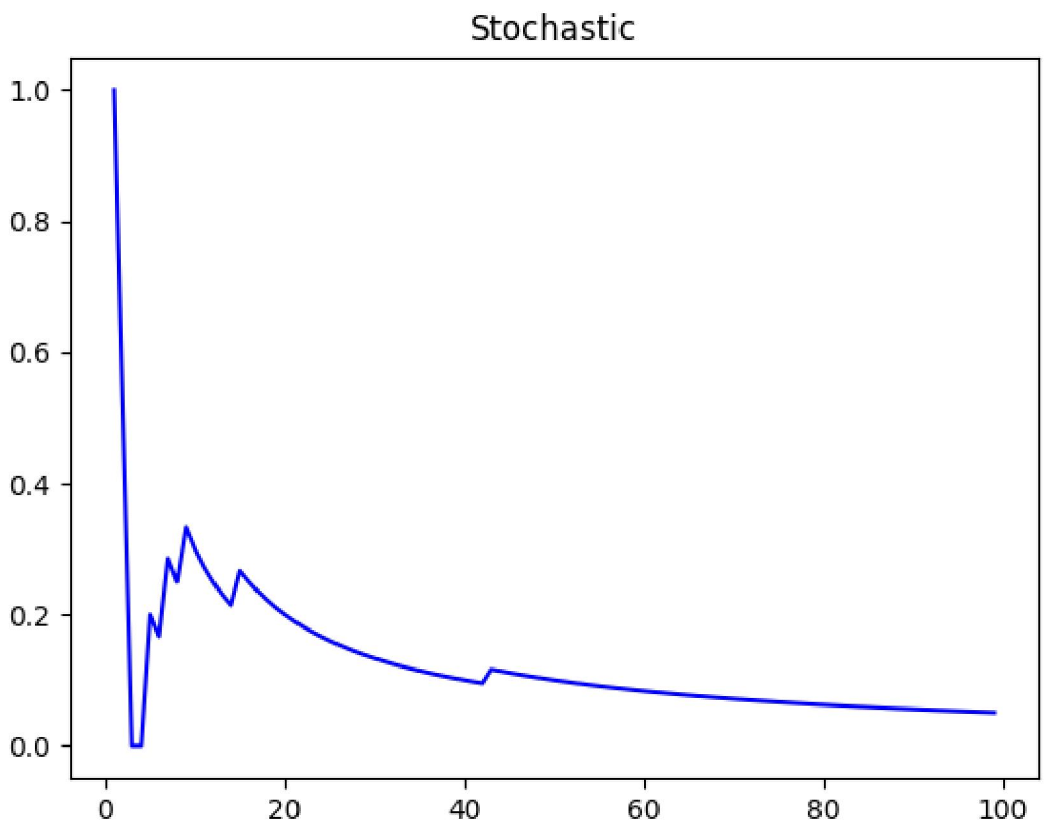


Figure 14: Weighted Majority Expert Losses Stochastic Graph using Extra Observations

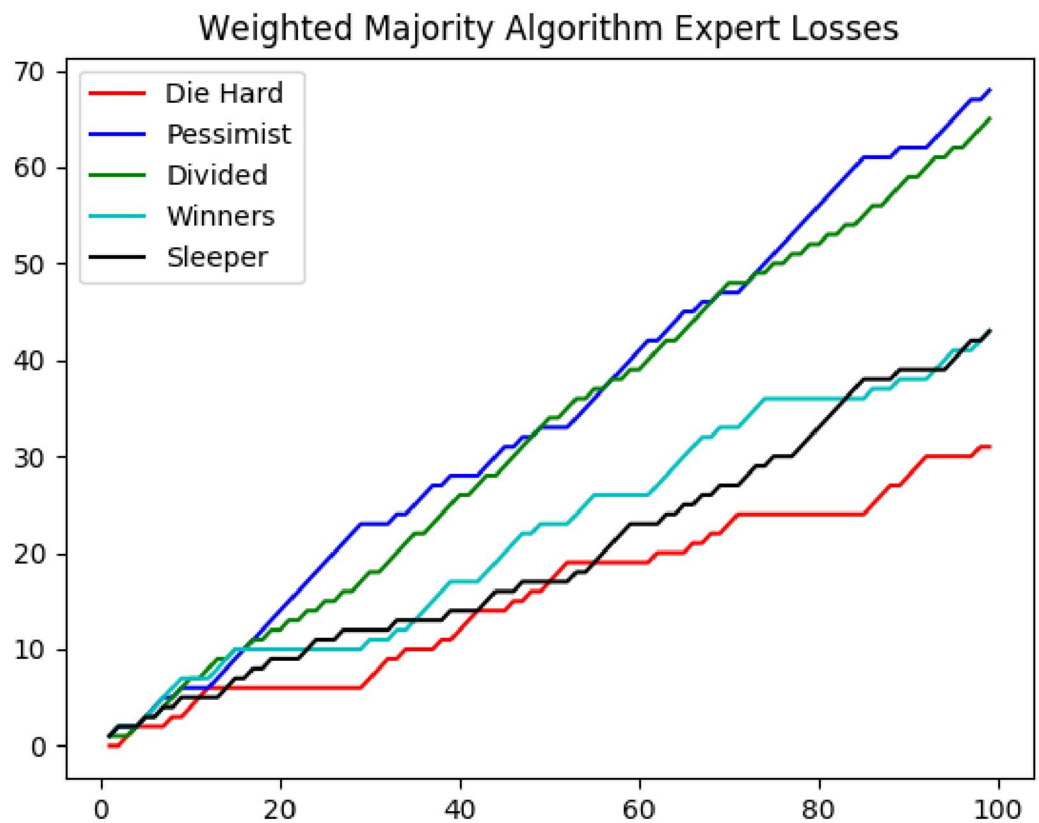


Figure 15: Weighted Majority Deterministic Graph using Extra Observations

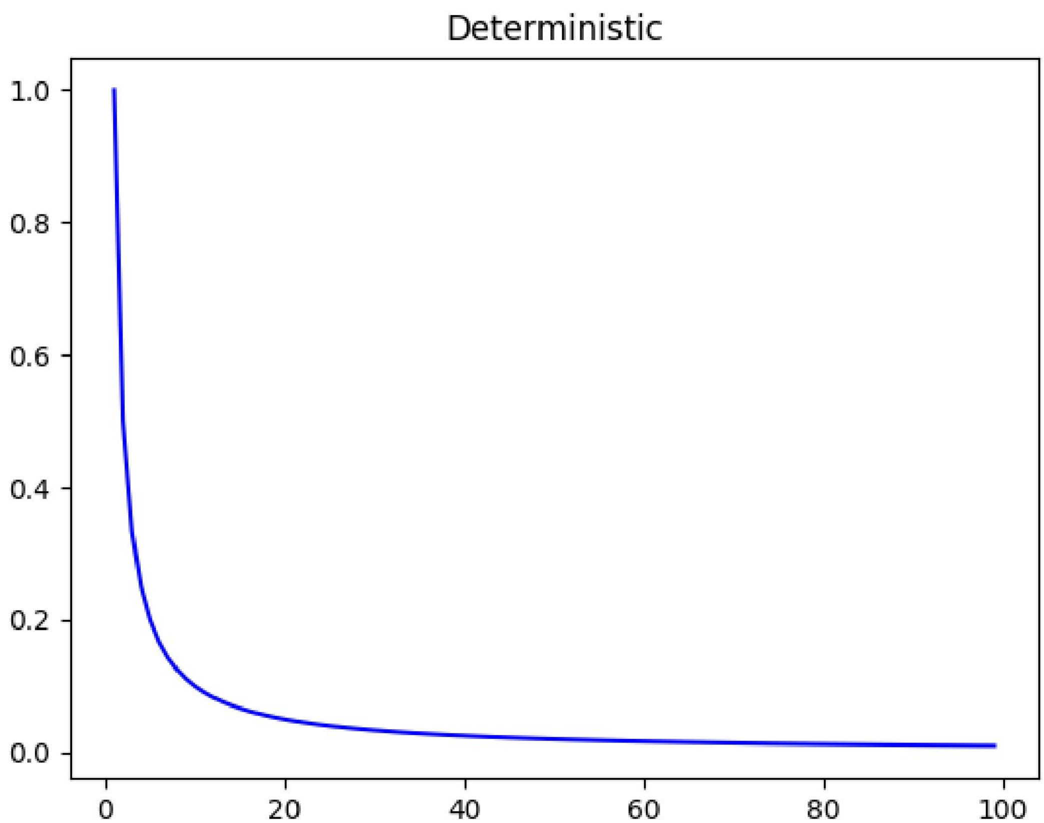


Figure 16: Weighted Majority Expert Losses Deterministic Graph using Extra Observations

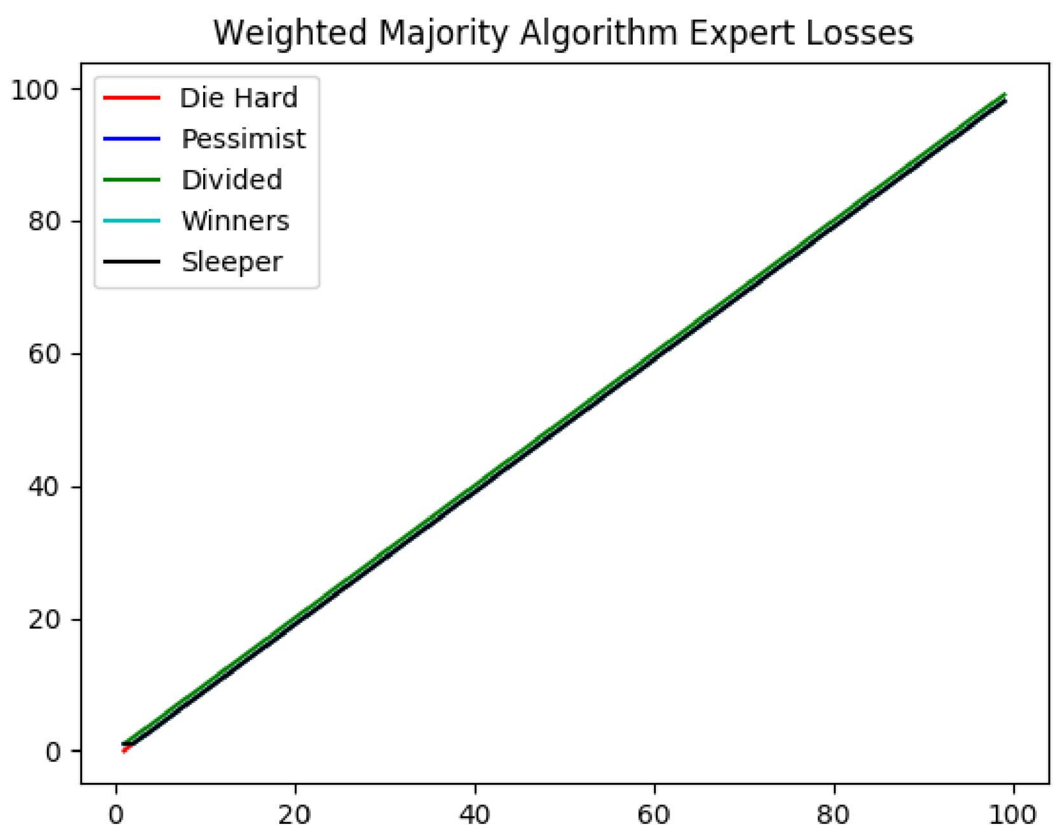


Figure 17: Weighted Majority Adversarial Graph using Extra Observations

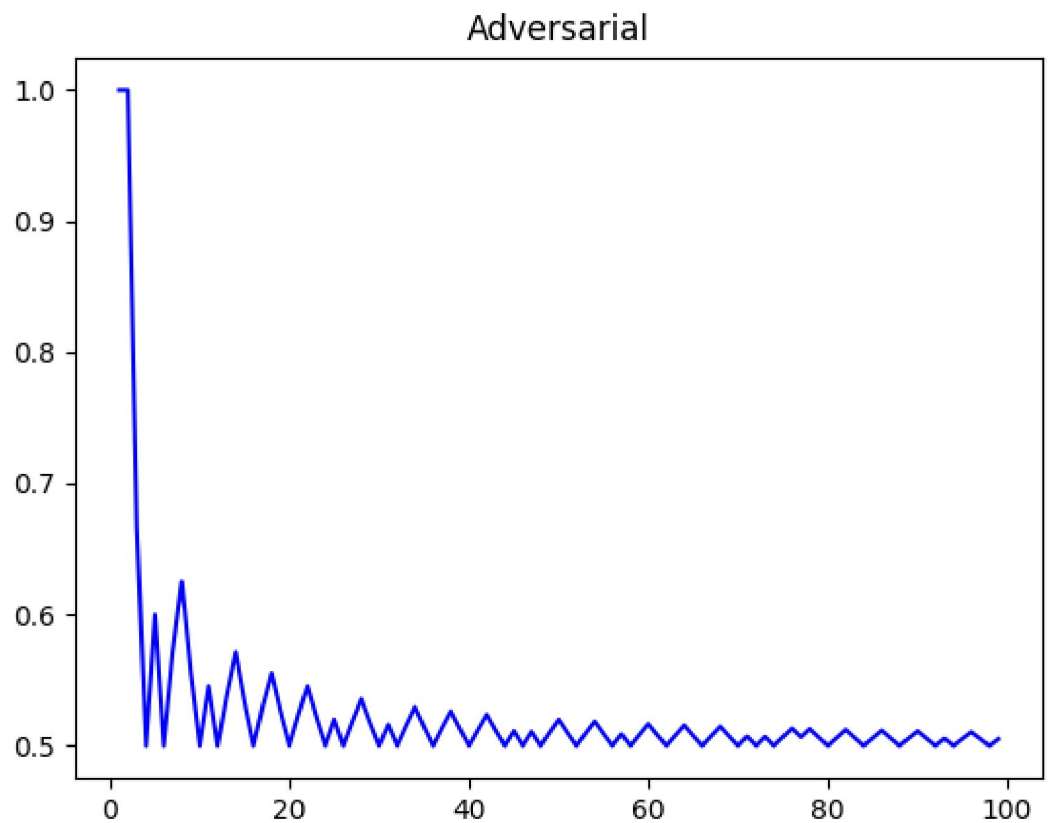


Figure 18: Weighted Majority Expert Losses Adversarial Graph using Extra Observations

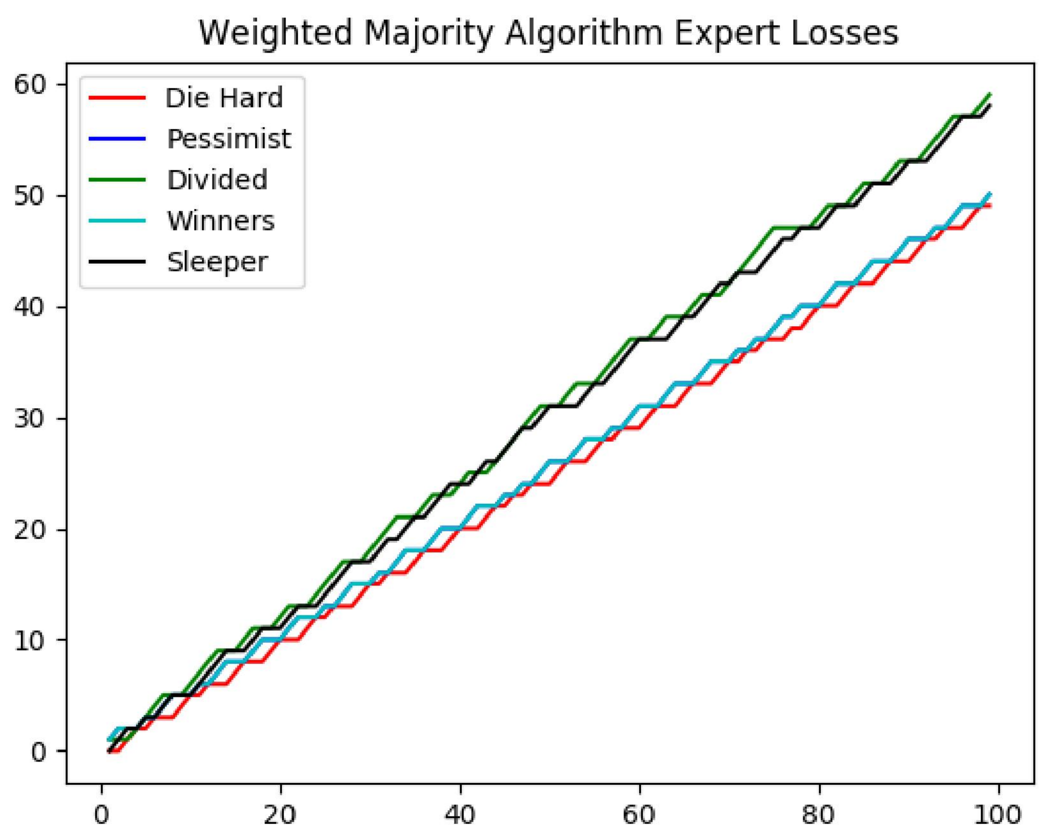
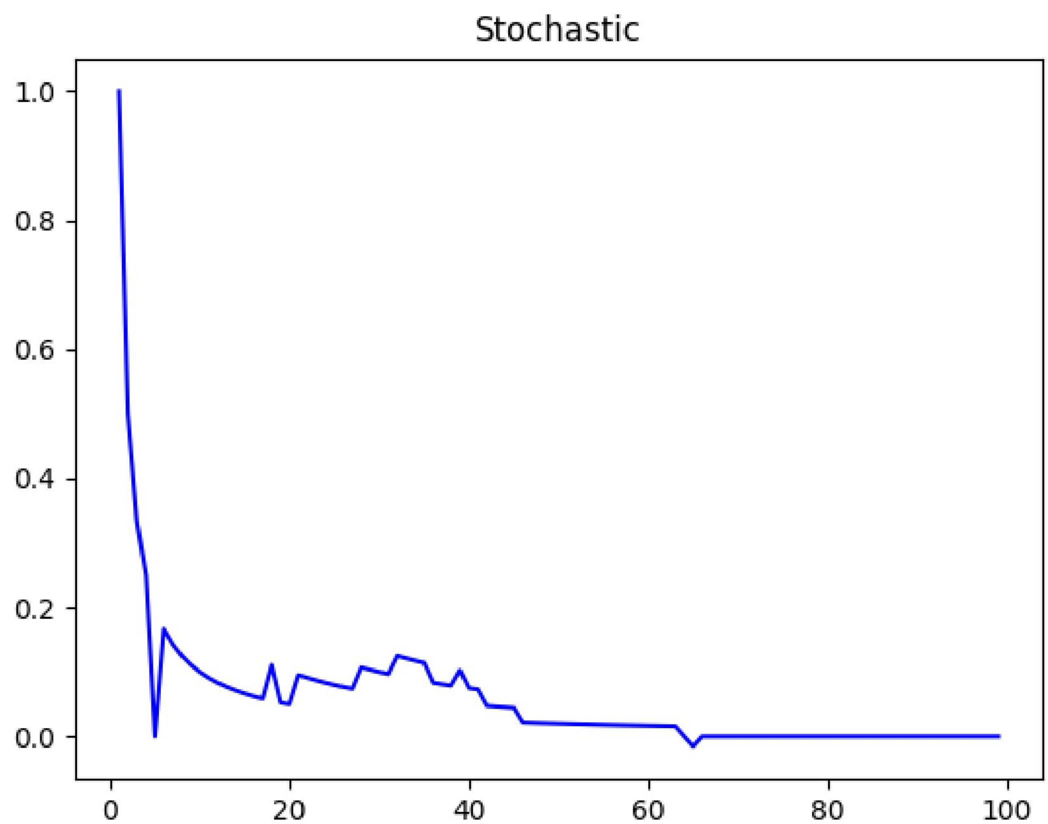


Figure 19: Random Weighted Majority Stochastic Graph using Extra Observations



The stochastic plot is determined by a random selection of whether the students go sleep and if they had tests or not. It is also influenced by how many games they won recently and the ratio of games they won overall.

Figure 20: Random Weighted Majority Stochastic Expert Loss Graph using Extra Observations

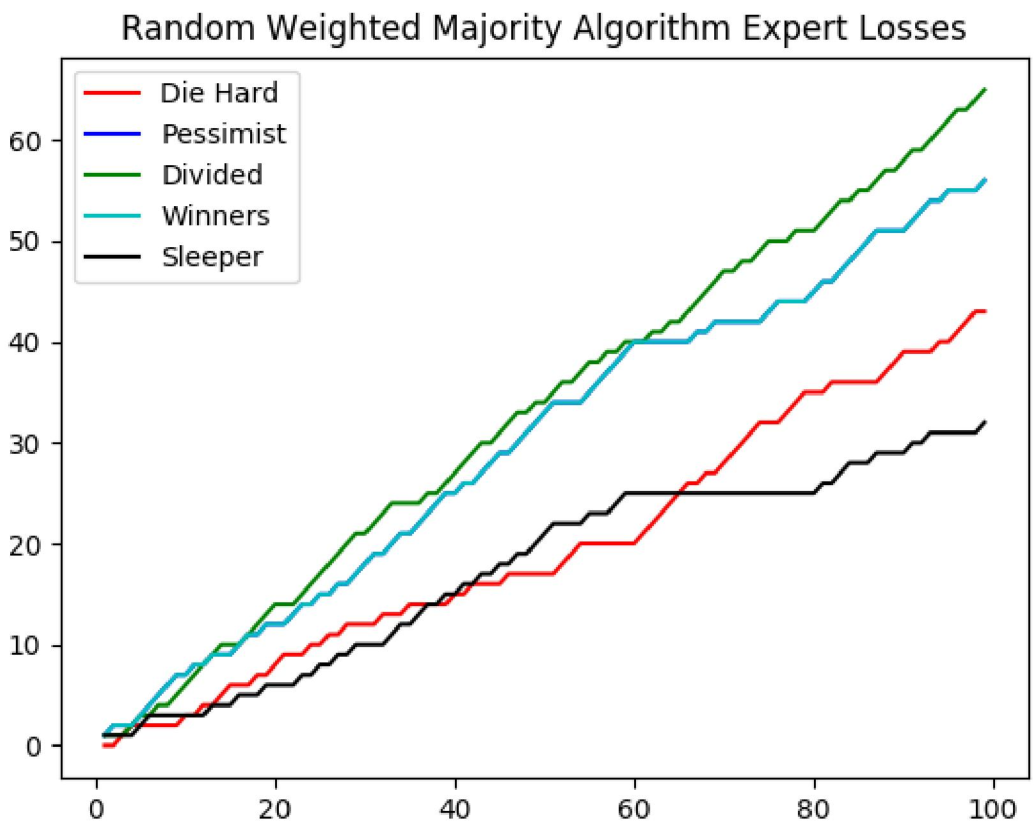
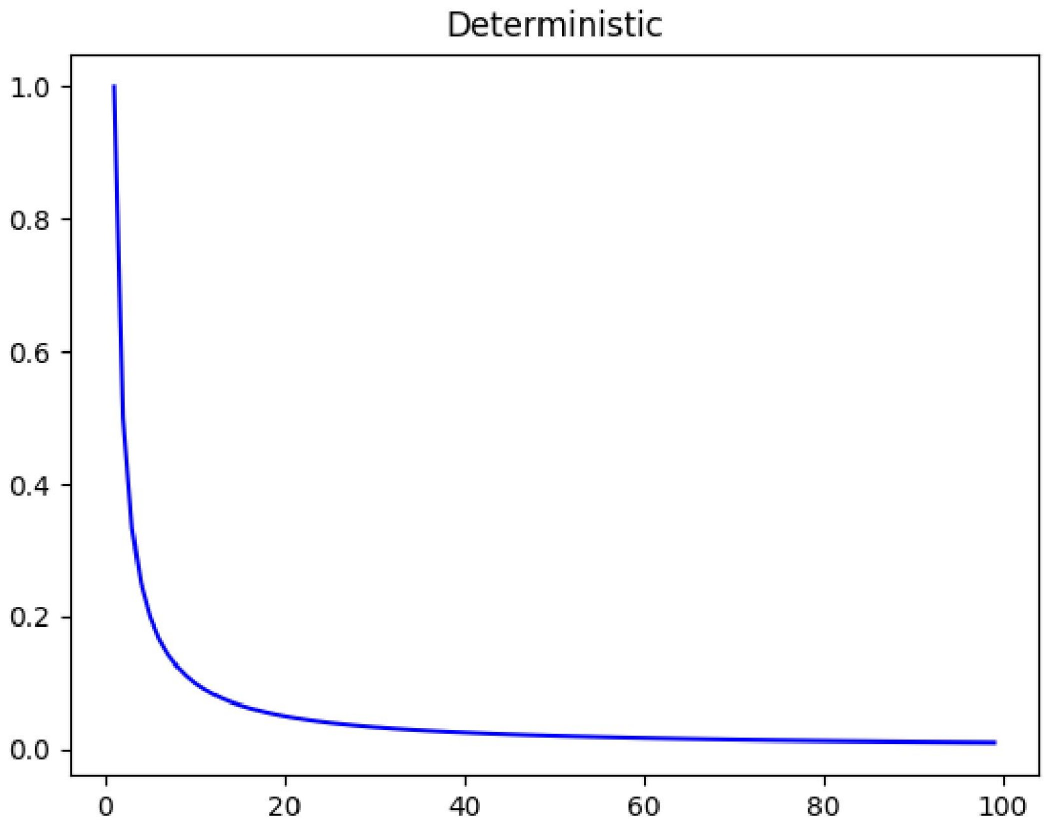


Figure 21: Random Weighted Majority Deterministic Graph using Extra Observations



The deterministic plot was influenced by if the sum of the the experts and weights were even or not. The new observations and weights were factored into this decision.

Figure 22: Random Weighted Majority Deterministic Expert Loss Graph using Extra Observations

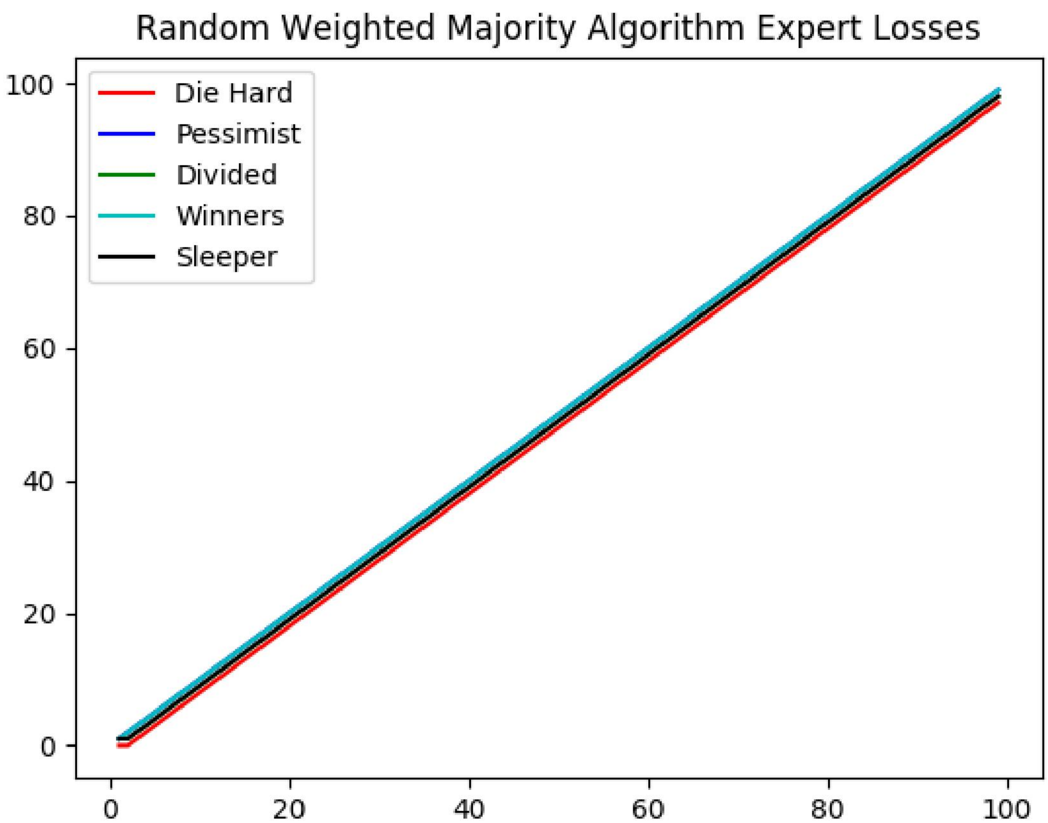
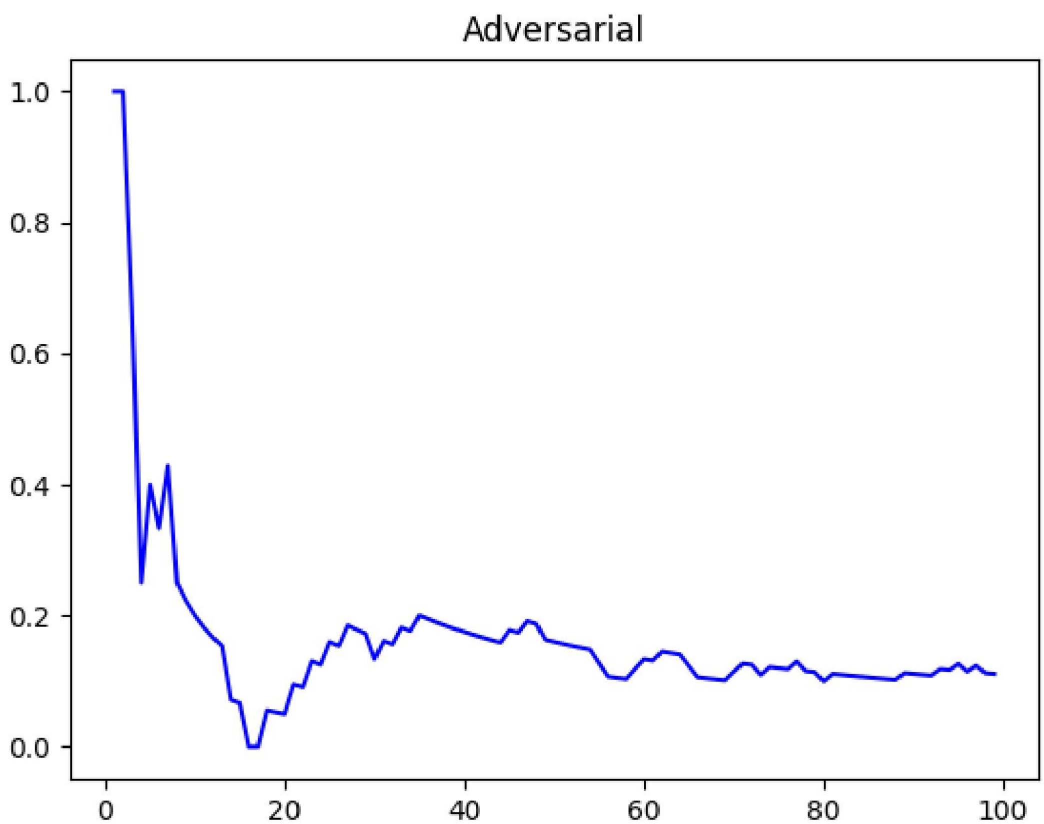
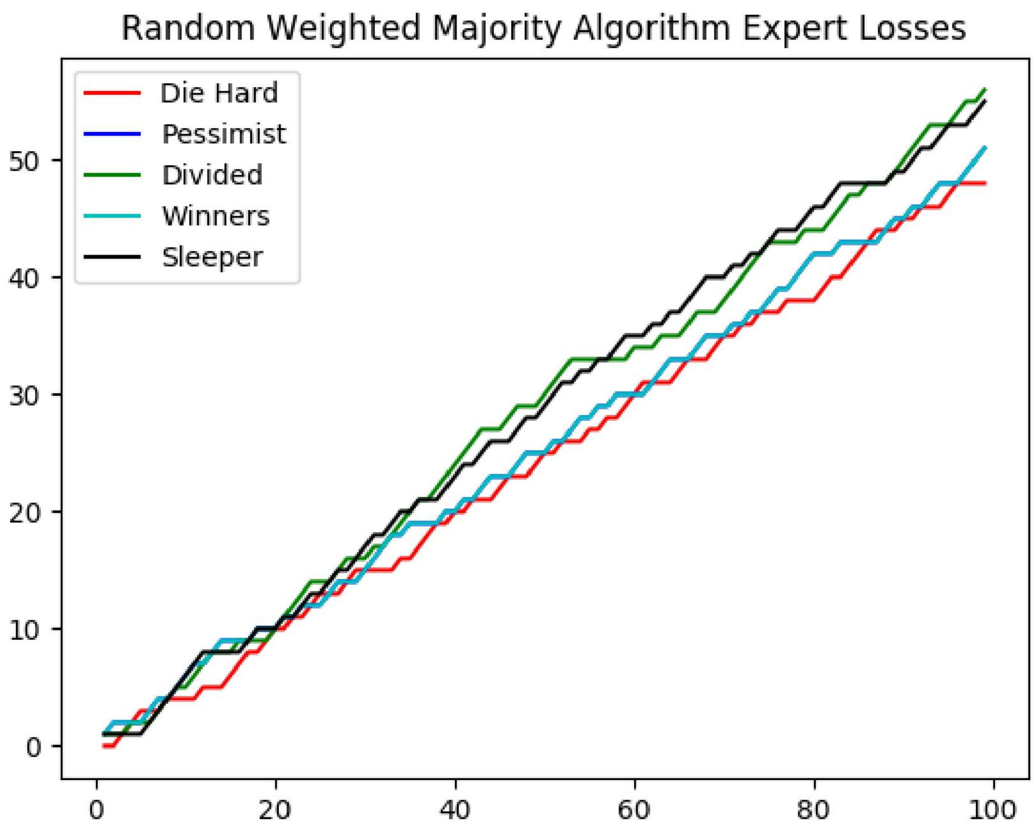


Figure 23: Random Weighted Majority Adversarial Graph using Extra Observations



The adversarial plot was influenced by the random observations, which helped the graph converge faster with less oscillation.

Figure 24: Random Weighted Majority Adversarial Expert Loss Graph using Extra Observations



3.4 More experts and observations/features 15 / 15

✓ - 0 pts Correct

gradescope