CSCE 658: Randomized Algorithms

Lecture 17

Samson Zhou

Relevant Supplementary Material

- Lecture 13 of "Advanced Algorithms" Course Notes (http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15850-f20/www/notes/lec14.pdf), by Anupam Gupta

Online Learning

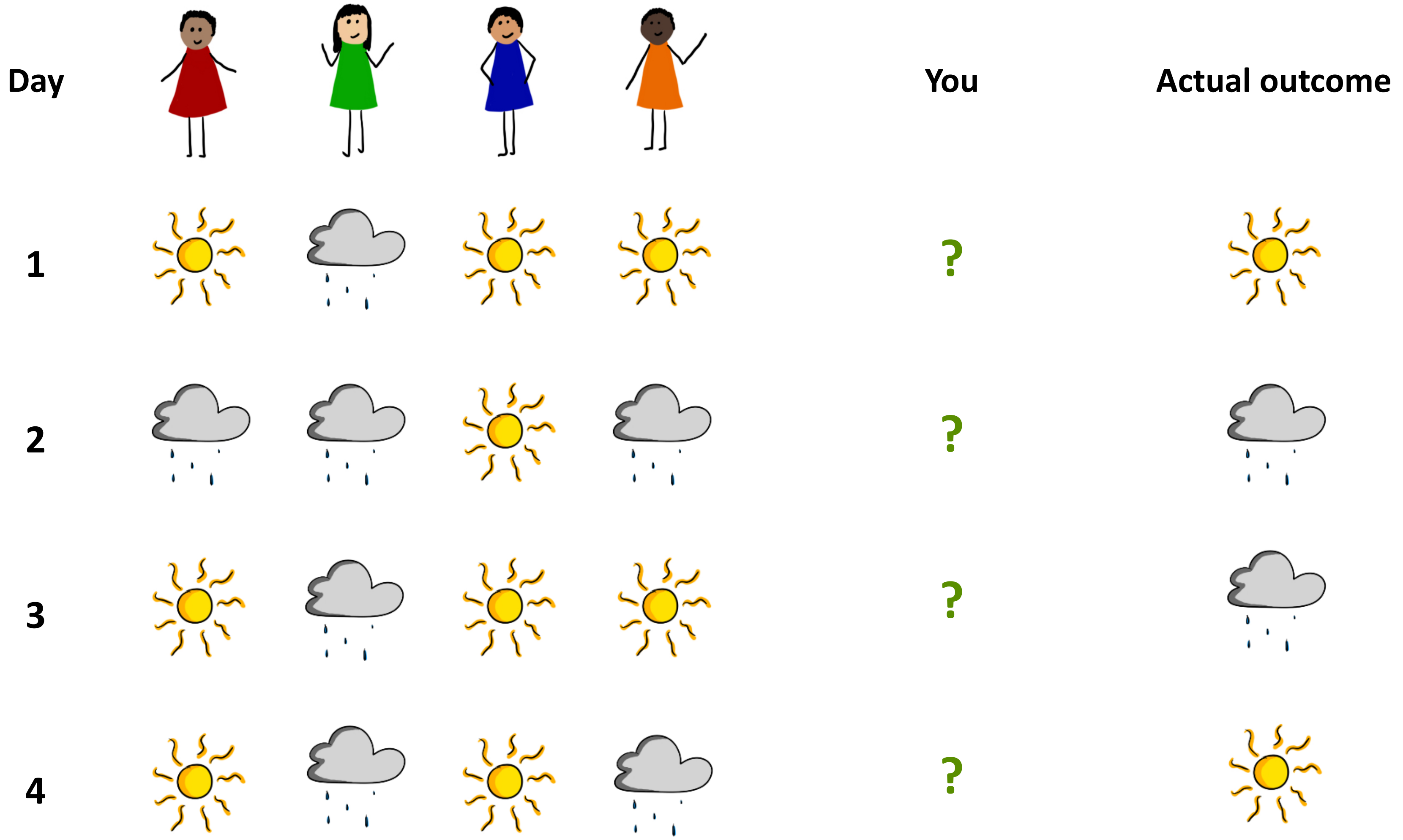
- There are n experts who make a prediction about each of T days ($n \gg T$)

- Algorithm uses advice from experts to make predictions each day

- Goal is to minimize the number of mistakes, i.e., the number of times our prediction differs from the outcome

Prediction with Expert Advice

a fundamental problem of **sequential prediction**

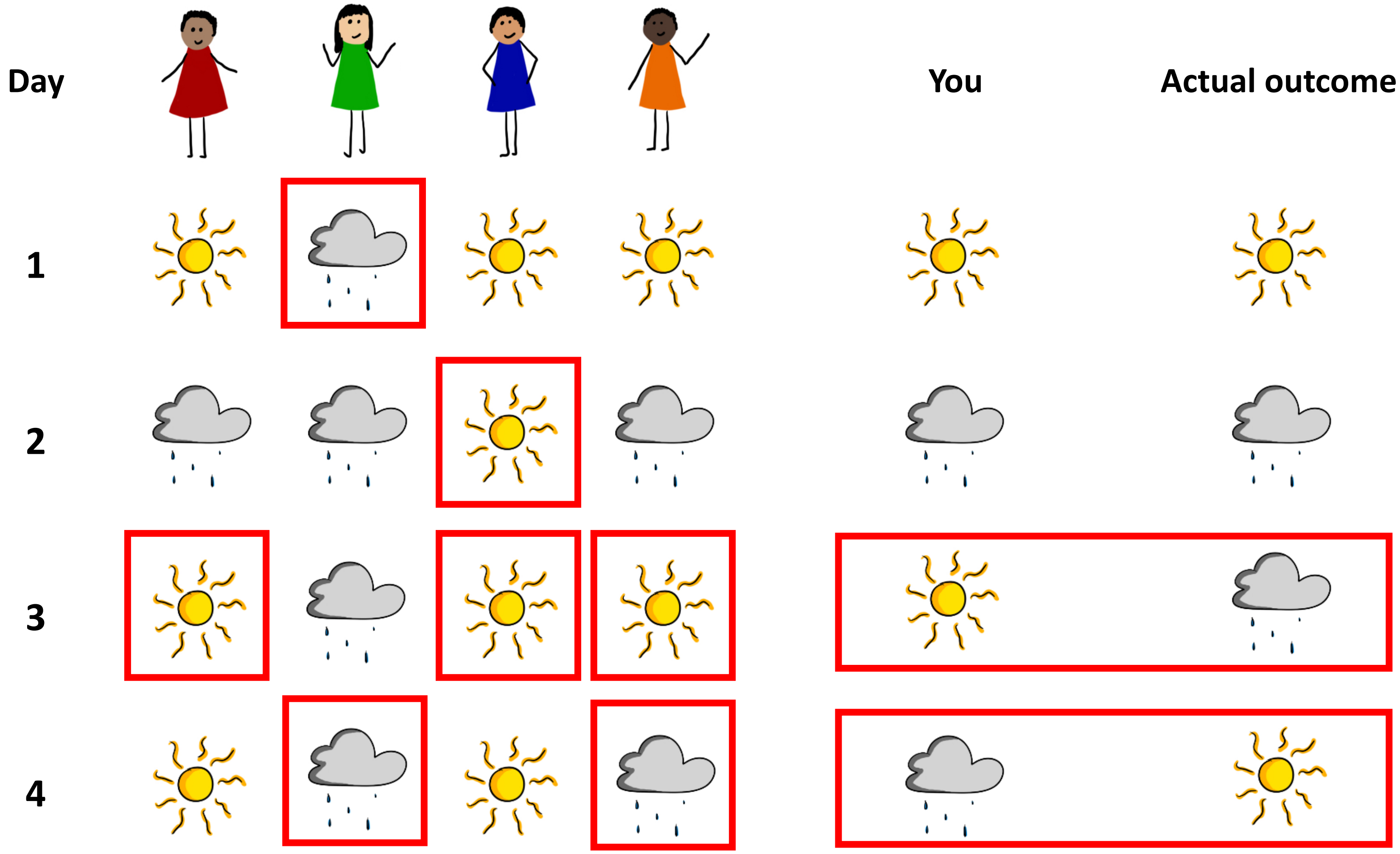


The Online Learning with Experts Problem

- n experts who decide either $\{0,1\}$ on each of T days ($n \gg T$)
- Algorithm takes advice from experts and predict either $\{0,1\}$ on each day
- Algorithm sees the outcome, which is either $\{0,1\}$, of each day and can use this information on future days
- The cost of the algorithm is the number of incorrect predictions

Prediction with Expert Advice

a fundamental problem of **sequential prediction**



Algorithm makes 2 mistakes

Best expert makes 1 mistake

Applications of the Experts Problem

- Ensemble learning, e.g., AdaBoost

- Forecast and portfolio optimization

- Special case of online convex optimization

Perfect Expert

- Theorem: If there is a perfect expert, there exists an algorithm that makes at most $\lceil \log_2 n \rceil$ mistakes

- Consider majority vote of all experts who have made no mistakes so far

- Every time we make a mistake, the number of mistakes who have not been wrong decreases by a factor of at least 2

Errors by Algorithm

- Theorem: Any algorithm MUST make at least $\lceil \log_2 n \rceil$ mistakes

- Suppose on day i, the experts with i-th bit 0 in their binary representation predict 0 and the experts with i-th bit 1 in their binary representation predict 1

Algorithms for Online Learning

- Theorem: There exists an algorithm that makes $M \leq m^*(\lceil \log_2 n \rceil + 1) + \lceil \log_2 n \rceil$ mistakes, where m^* is the number of mistakes made by the best expert

Algorithms for Online Learning

- Theorem: There exists an algorithm that makes $M \leq m^*(\lceil \log_2 n \rceil + 1) + \lceil \log_2 n \rceil$ mistakes, where m^* is the number of mistakes made by the best expert

- Split the time into epochs
- Keep perfect experts in each epoch and do majority vote
- When no more perfect experts, epoch ends and start a new epoch with all experts

Algorithms for Online Learning

- Theorem: There exists an algorithm that makes $M \leq m^*(\lceil \log_2 n \rceil + 1) + \lceil \log_2 n \rceil$ mistakes, where m^* is the number of mistakes made by the best expert

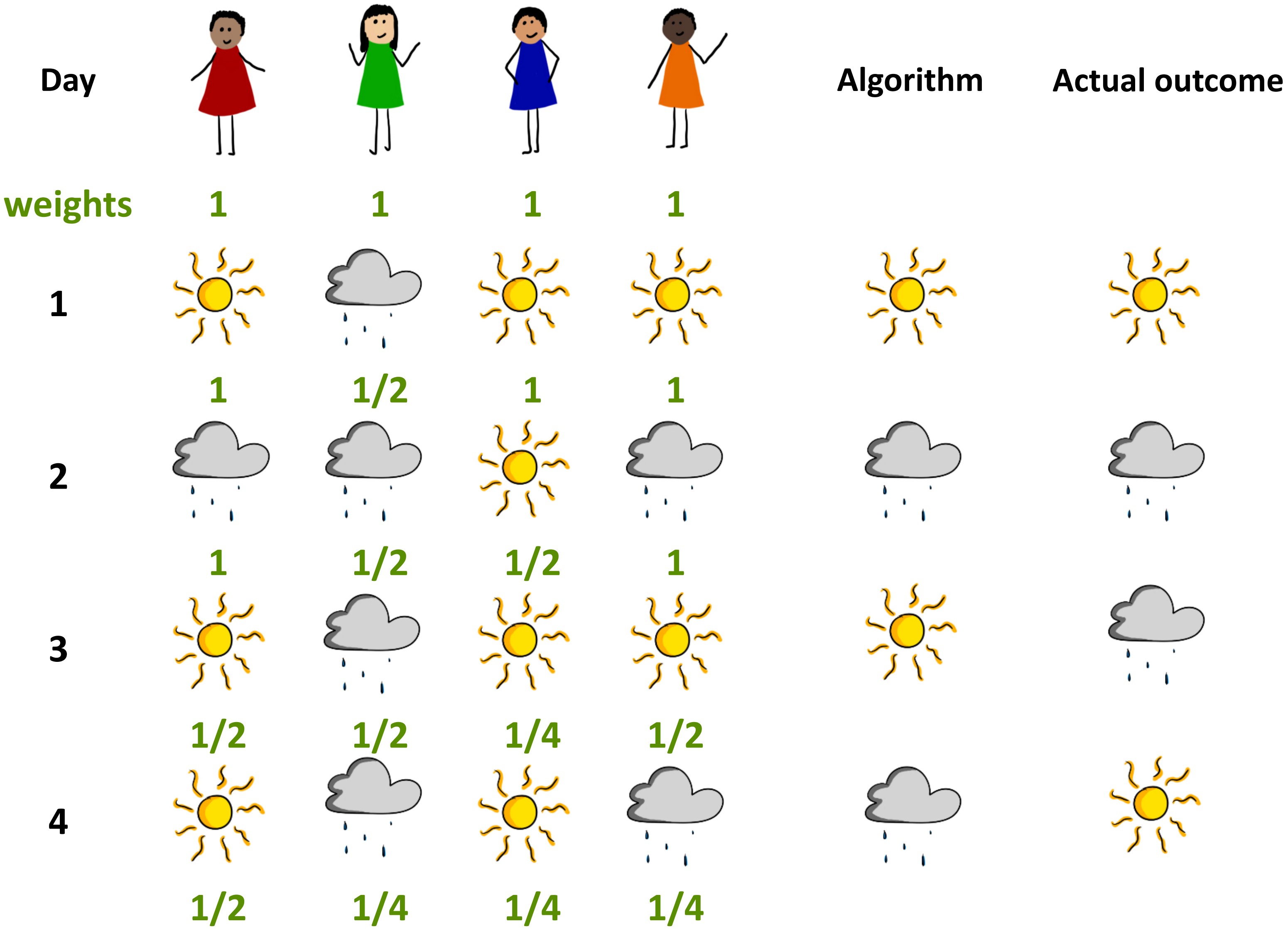
- Split the time into epochs
- Keep perfect experts in each epoch and do majority vote
- When no more perfect experts, epoch ends and start a new epoch with all experts HOW MANY EPOCHS CAN THERE BE?

Algorithms for Online Learning

- Theorem: There exists an algorithm that makes $M \leq m^*(\lceil \log_2 n \rceil + 1) + \lceil \log_2 n \rceil$ mistakes, where m^* is the number of mistakes made by the best expert

- $\lceil \log_2 n \rceil + 1$ mistakes per epoch before there is a perfect expert
- $\lceil \log_2 n \rceil$ mistakes when there is a perfect expert
- m^* epochs before there is a perfect expert

Weighted Majority (Littlestone, Warmuth 89)



Weighted Majority (Littlestone, Warmuth 89)

- Initially give each expert weight 1
- Choose the weighted majority of experts
- For each time, maintain the weight of each correct expert, decrease the weight of each incorrect expert by $\frac{1}{2}$

Guarantee for Weighted Majority

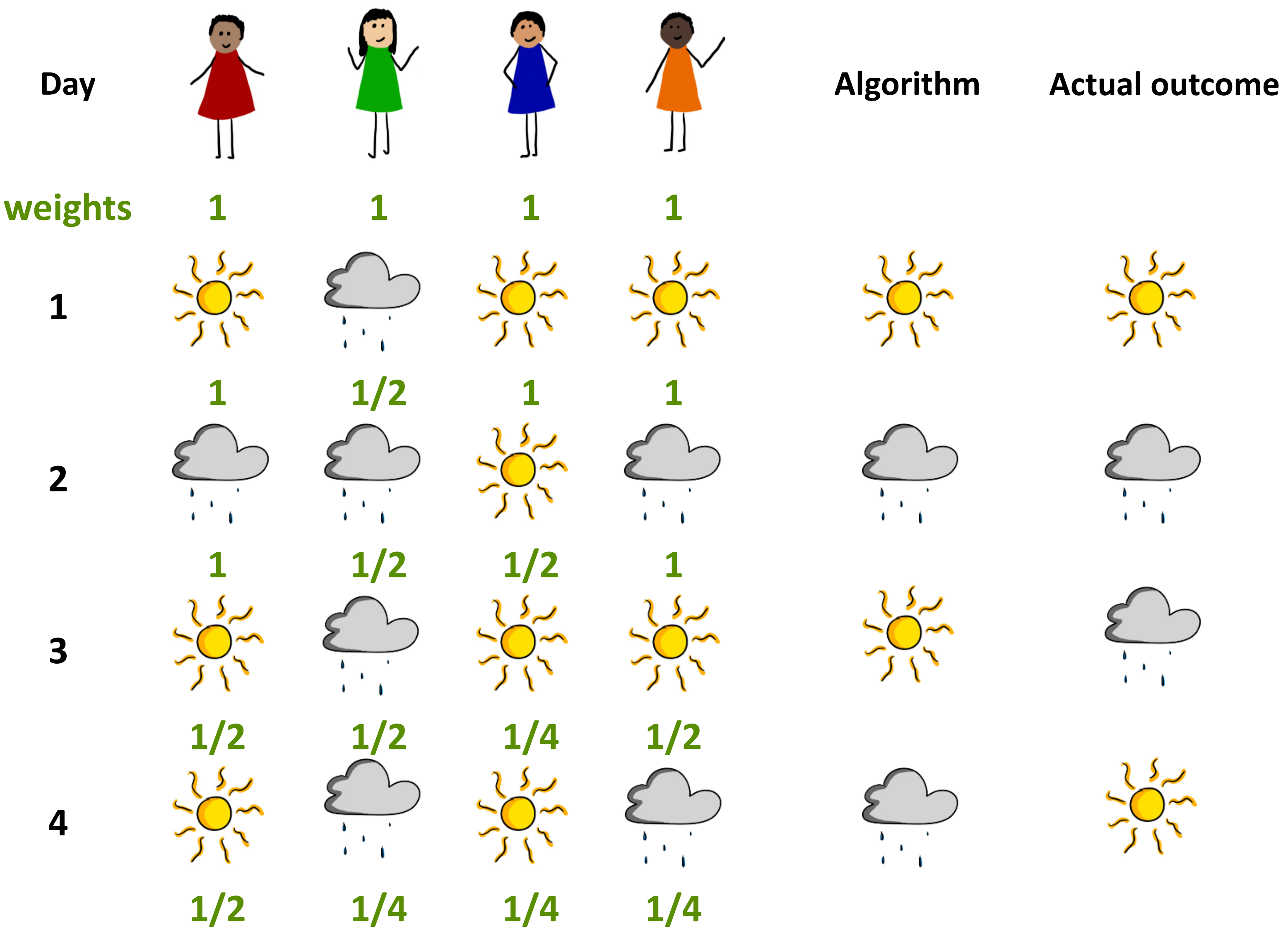
- What is the sum of the weights at the beginning? n
- What is an upper bound on the weights in each round?
- Each round the algorithm makes a mistake, at least half of its experts have their weights decrease by half
- Sum of the weights $\leq \left(1 - \frac{1}{4}\right)^M n$

Guarantee for Weighted Majority

- What is the weight of the best expert at the end? $\frac{1}{2^{m^*}}$
- Sum of the weights $\geq \frac{1}{2^{m^*}}$

- $\frac{1}{2^{m^*}} \leq$ sum of the weights $\leq \left(1 - \frac{1}{4}\right)^M n$
- $M \leq \frac{m^* + \log_2 n}{\log_2 \frac{4}{3}} \approx 2.41(m^* + \log_2 n)$

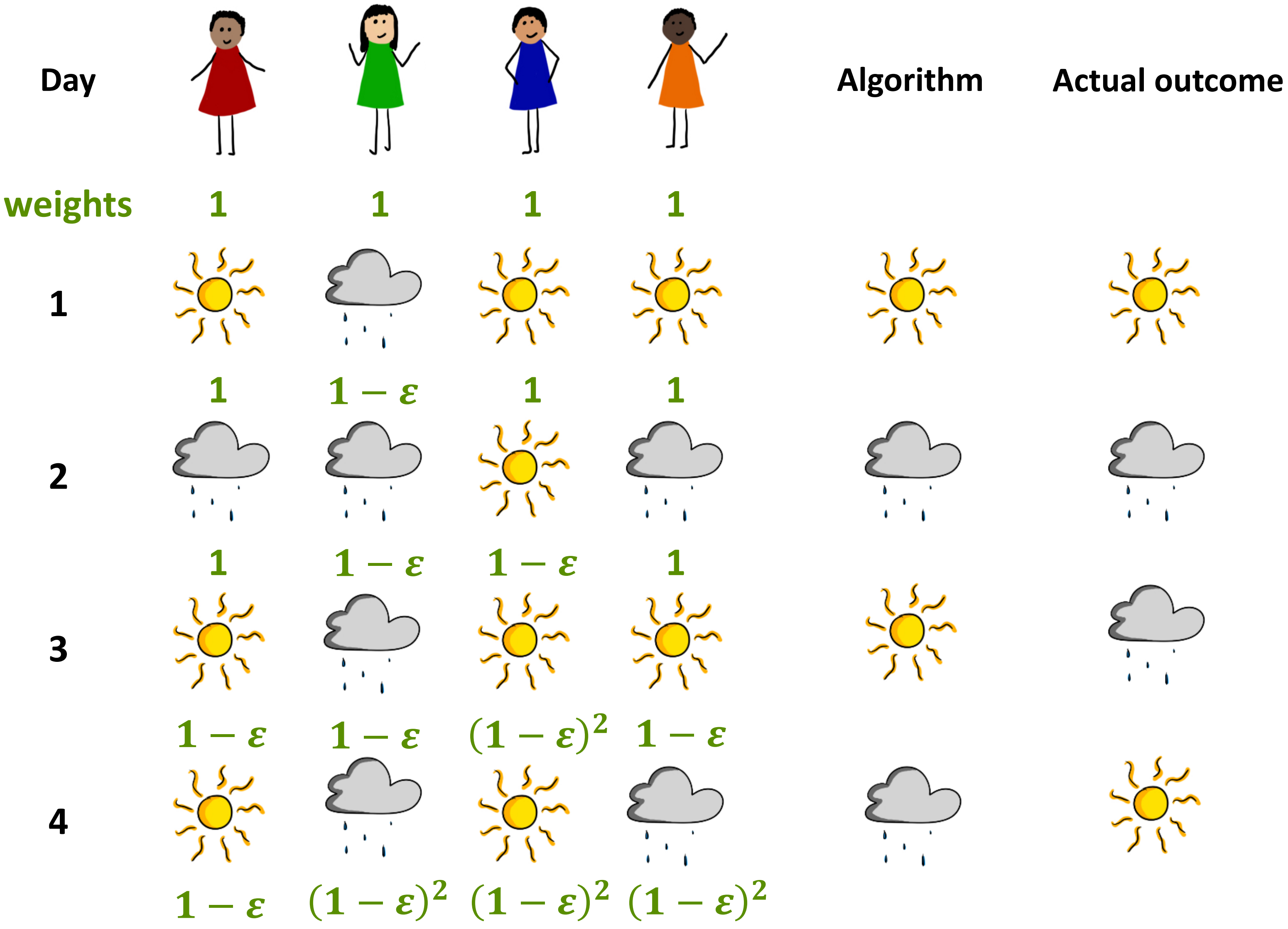
Weighted Majority (Littlestone, Warmuth 89)



Weighted Majority (Littlestone, Warmuth 89)

- Initially give each expert weight 1
- Choose the weighted majority of experts
- For each time, maintain the weight of each correct expert, decrease the weight of each incorrect expert by $(1 - \varepsilon)$

Weighted Majority (Littlestone, Warmuth 89)



Guarantee for Weighted Majority

- What is the sum of the weights at the beginning? n
- What is an upper bound on the weights in each round?
- Each round the algorithm makes a mistake, at least half of its experts have their weights decrease by $(1-\varepsilon)$
- Sum of the weights $\leq \left(1-\frac{\varepsilon}{2}\right)^{M} n$

Guarantee for Weighted Majority

- What is the weight of the best expert at the end? $(1-\varepsilon)^{m^*}$
- Sum of the weights $\geq (1-\varepsilon)^{m^*}$

- $(1-\varepsilon)^{m^*} \leq$ sum of the weights $\leq \left(1-\frac{\varepsilon}{2}\right)^{M} n \leq e^{-\frac{\varepsilon M}{2}} n$
- $M \leq 2(1+\varepsilon)m^* + O\left(\frac{\log n}{\varepsilon}\right)$
- (since $-\ln(1-\varepsilon) = \varepsilon + \frac{\varepsilon^2}{2} + \frac{\varepsilon^3}{3} + \cdots \leq \varepsilon + \varepsilon^2$ for $\varepsilon \in [0,1]$)

Deterministic Algorithm Error

- Theorem: No deterministic algorithm can do better than a factor of 2 compared to the best expert

- Consider two experts A and B, where A always picks 1 and B always picks 0
- Since the algorithm is deterministic, can always make the algorithm wrong on the next day
- Algorithm is incorrect every day, some expert is correct on half of the days

Randomized Weighted Majority (Littlestone, Warmuth 89)

- Initially give each expert weight 1
- Prediction at each time is drawn randomly proportional to the current weights of the experts
- For each time, maintain the weight of each correct expert, decrease the weight of each incorrect expert by $(1 - \varepsilon)$

Randomized Weighted Majority (Littlestone, Warmuth 89)

- Let the potential function Φ_t denote the sum of the weights at time t

- Let f_t be the fraction of incorrect experts at time t
- By linearity of expectation $E[M] = \sum_t f_t$
- We have $\Phi_{t+1} = \Phi_t\big((1 - f_t) + f_t(1 - \varepsilon)\big) = \Phi_t(1 - \varepsilon f_t)$
- $\Phi_T = n \prod_t (1 - \varepsilon f_t) \leq ne^{-\varepsilon \sum_t f_t} = ne^{-\varepsilon E[M]}$

Randomized Weighted Majority (Littlestone, Warmuth 89)

- We have $\Phi_T \leq ne^{-\varepsilon E[M]}$
- We also have $(1-\varepsilon)^{m^*} \leq \Phi_T$
- Then $E[M] \leq m^*(1+\varepsilon) + \frac{\ln n}{\varepsilon}$ by using $-\ln(1-\varepsilon) \leq \varepsilon + \varepsilon^2$

Randomized Weighted Majority (Littlestone, Warmuth 89)

- Initially give each expert weight 1

- Prediction at each time is drawn randomly proportional to the current weights of the experts

- For each time, maintain the weight of each correct expert, decrease the weight of each incorrect expert by $(1-\varepsilon)$

Multiplicative Weights

- Initially give each expert weight 1
- Prediction at each time is drawn randomly proportional to the current weights of the experts
- For each time t, change the weight of each expert by $(1 - \varepsilon m_i^{(t)})$, where $m_i^{(t)}$ is the loss of the i-th expert

An Alternate Perspective

- Suppose in each round that the algorithm produces a vector of probabilities $p^t = (p_1^t, \ldots, p_n^t)$

- We have $p_i^t \in [0,1]$ for all $i \in [n]$ and $p_1^t + \cdots + p_n^t = 1$

- p_i^t corresponds to the probability of picking expert i on day t

An Alternate Perspective

- The loss on day t is $\ell^t = (\ell_1^t, \dots, \ell_n^t) \in [0,1]^n$
- p_i^t corresponds to the probability of picking expert i on day t
- Expected loss on day t is $\langle p^t, \ell^t \rangle$

Hedge Algorithm

- Initially give each expert weight 1
- On day t, randomly follows expert i with probability $p_i^t = \frac{w_i^t}{w_1^t + \cdots + w_n^t} = \frac{w_i^t}{\sum_j w_j^t}$
- Each weight is updated by $w_i^{t+1} = w_i^t \cdot \exp(-\varepsilon \ell_i^t)$

Hedge Algorithm

- Let the potential function Φ_t denote the sum of the weights at time t
- We have: $$\Phi_{t+1} = \sum_i w_i^{t+1} = \sum_t w_i^t e^{-\varepsilon \ell_i^t}$$
- Since $e^x \leq 1 + x + x^2$ for $x \in [-1,1]$ and also $\left|\ell_i^t\right| \leq 1$, we have
$$\Phi_{t+1} \leq \sum_i w_i^t \left(1 - \varepsilon \ell_i^t + \varepsilon^2 \left(\ell_i^t\right)^2\right)$$
$$\leq \sum_i w_i^t (1 + \varepsilon^2) - \varepsilon \sum_i w_i^t \ell_i^t$$

Hedge Algorithm

- We have $\Phi_{t+1} \leq \sum_i w_i^t (1+\varepsilon^2) - \varepsilon \sum_i w_i^t \ell_i^t$

- Since $w_i^t = p_i^t \cdot \Phi^t$, then

$$\Phi_{t+1} \leq (1+\varepsilon^2)\Phi^t - \varepsilon \Phi^t \langle p^t, \ell^t \rangle$$

$$= \Phi^t (1 + \varepsilon^2 - \varepsilon \langle p^t, \ell^t \rangle)$$

(since $1 + x \leq e^x$) $\leq \Phi^t \exp(\varepsilon^2 - \varepsilon \langle p^t, \ell^t \rangle)$

Hedge Algorithm

- We have $\Phi_T \leq n \exp(\varepsilon^2 T - \varepsilon \sum_t \langle p^t, \ell^t \rangle)$
- We also have $\exp(-\varepsilon \sum_t \ell_{m^*}^t) \leq \Phi_T$
- Then $\sum_t \langle p^t, \ell^t \rangle \leq \sum_t \ell_{m^*}^t + \varepsilon T + \frac{\ln n}{\varepsilon}$
- If we set $\varepsilon = \sqrt{\frac{\ln n}{T}}$, we have $\varepsilon T + \frac{\ln n}{\varepsilon} = 2\sqrt{T \ln N}$

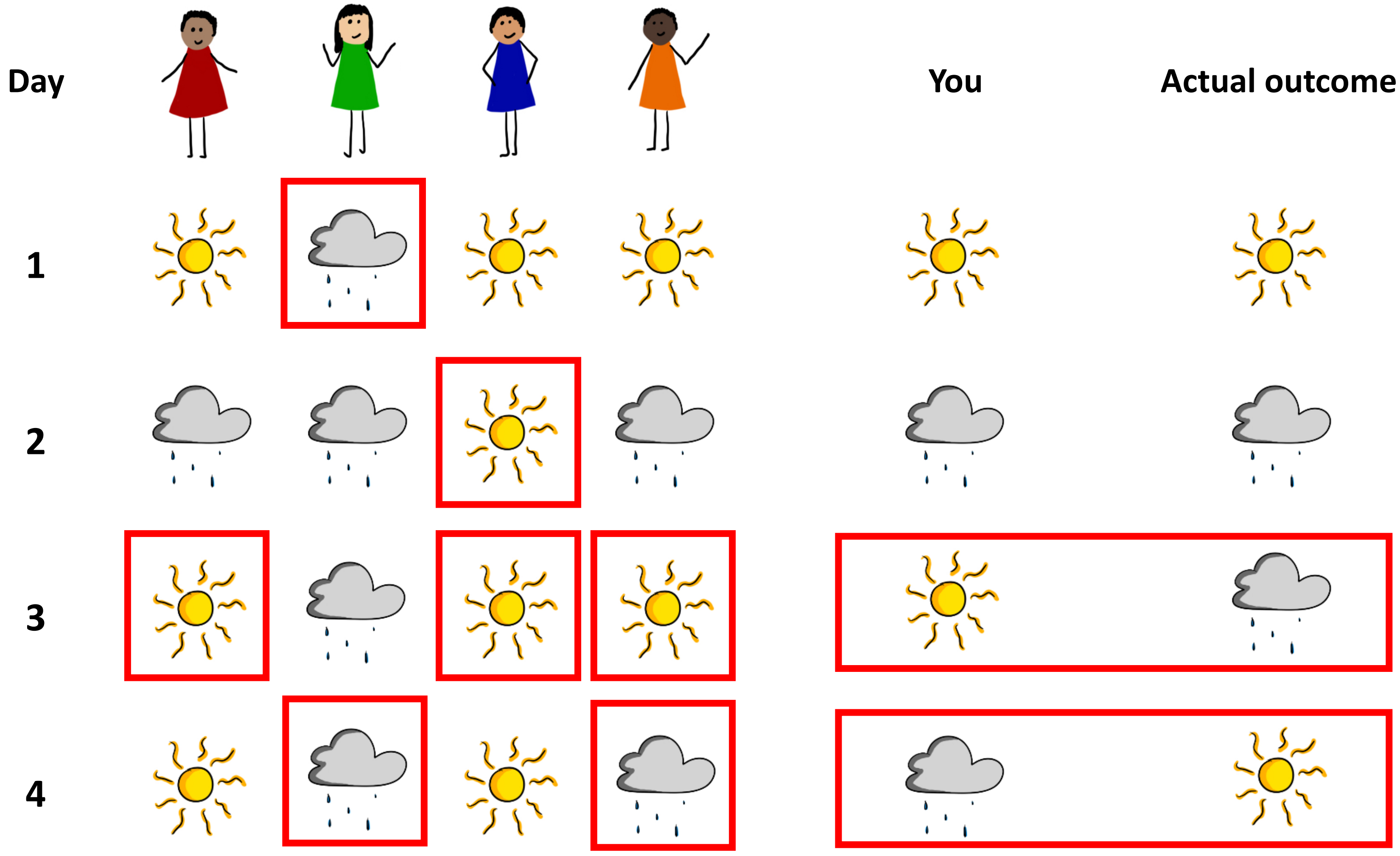
Regret

- Regret is the difference $M - m$ between the number of mistakes M by our algorithm and the number of mistakes by the best expert m^*

- (Amortized) regret is the ratio $\frac{M-m^*}{T}$, i.e., the regret amortized over the total number of days T

Prediction with Expert Advice

a fundamental problem of **sequential prediction**

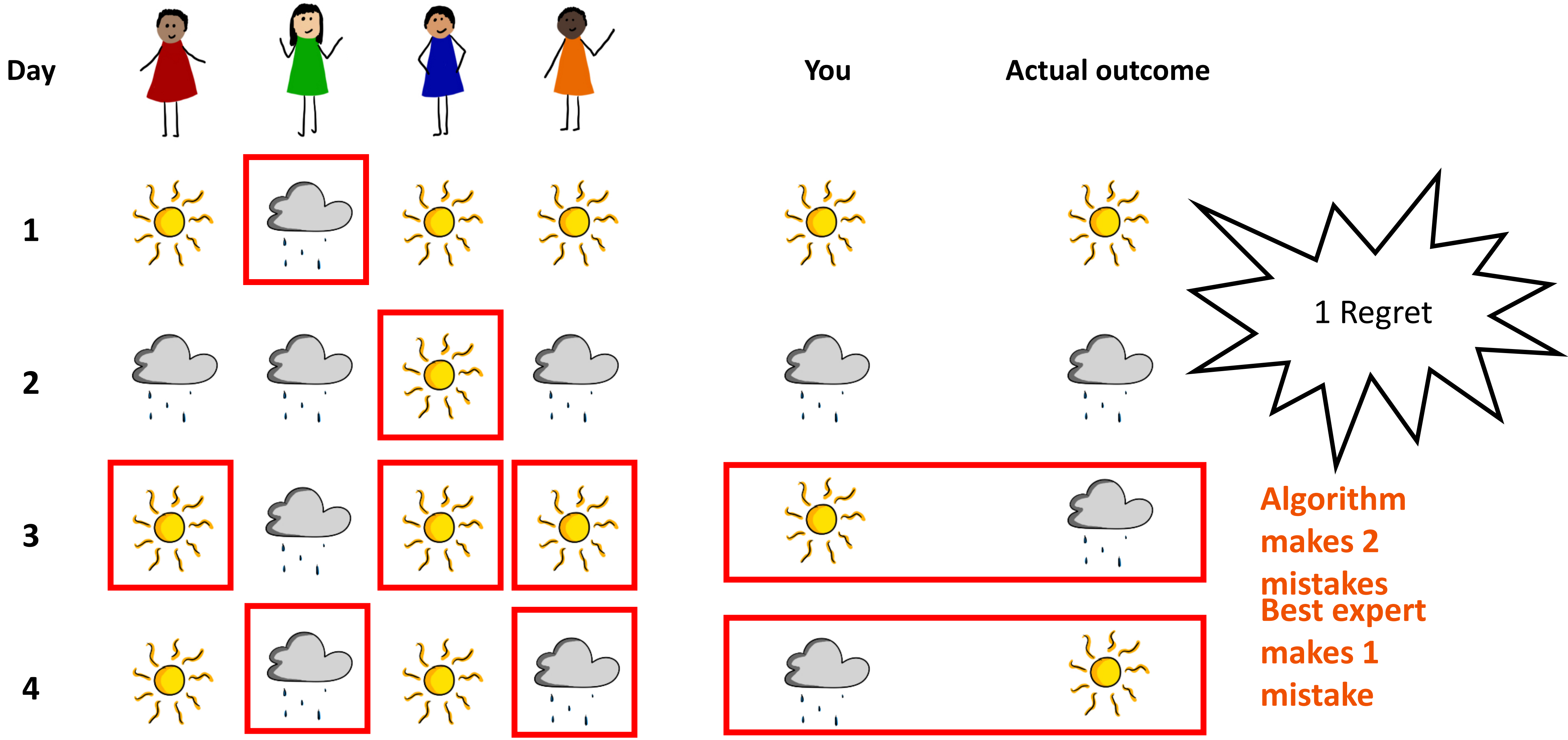


Algorithm makes 2 mistakes

Best expert makes 1 mistake

Prediction with Expert Advice

a fundamental problem of **sequential prediction**



Algorithm makes 2 mistakes
Best expert makes 1 mistake

Hedge Algorithm

- If we set $\varepsilon = \sqrt{\frac{\ln n}{T}}$, we have $\sum_t \langle p^t, \ell^t \rangle \leq \sum_t \ell_{m^*}^t + \varepsilon T + \frac{\ln n}{\varepsilon} \leq \sum_t \ell_{m^*}^t + 2\sqrt{T \ln N}$

- Total regret is $2\sqrt{T \ln N}$

- Amortized regret is $2\sqrt{\frac{\ln N}{T}}$