Normal-Hedge

The Hedge Algorithm

[Freund & Schapire 1997]

based on [Littlestone and Warmuth 1989, the Weighted Majority Algorithm]

Initial weights: $w^1 = \left\langle \frac{1}{N}, \ldots, \frac{1}{N} \right\rangle$

Weights update rule: $w_i^{t+1} = w_i^t e^{-\eta l_i^t}$

Learning rate

Alternatively: $w_i^{t+1} = \frac{1}{N} e^{-\eta L_i^t} \neq \frac{1}{N} \prod_{s=1}^{t} p_i^s\left(x^s\right)$

Pos

pro

(un-n

ood

probability

Not Bayes !

Potential-based bound

Potential: $W^t = \sum_{i=1}^{N} w_i^t$

Theorem: $\mathrm{L}_A^T \leq \dfrac{-\log W^{T+1}}{1-e^{-\eta}}$

Tuning the learning rate

$$\forall i,\ L_A^T \le \frac{\eta L_i^T + \ln N}{1 - e^{-\eta}}$$

If we set $\eta = \sqrt{\frac{2\ln N}{T}}$

Then we guarantee $\mathrm{L}_A^T \le \min_i L_i^T + \sqrt{2T\ln N} + \ln N$

Equivalently $\forall i, R_i^T \le \sqrt{2T\ln N} + \ln N; \quad \lim_{T\to\infty} \frac{\sqrt{2T\ln N} + \ln N}{T} = 0$

Achieved our goal!

Tuning the learning rate

$$\forall i,\ L_A^T \le \frac{\eta L_i^T + \ln N}{1 - e^{-\eta}}$$

If we set $\eta = \sqrt{\frac{2 \ln N}{T}}$

Then we guarantee $\mathrm{L}_A^T \le \min_i L_i^T + \sqrt{2T \ln N} + \ln N$

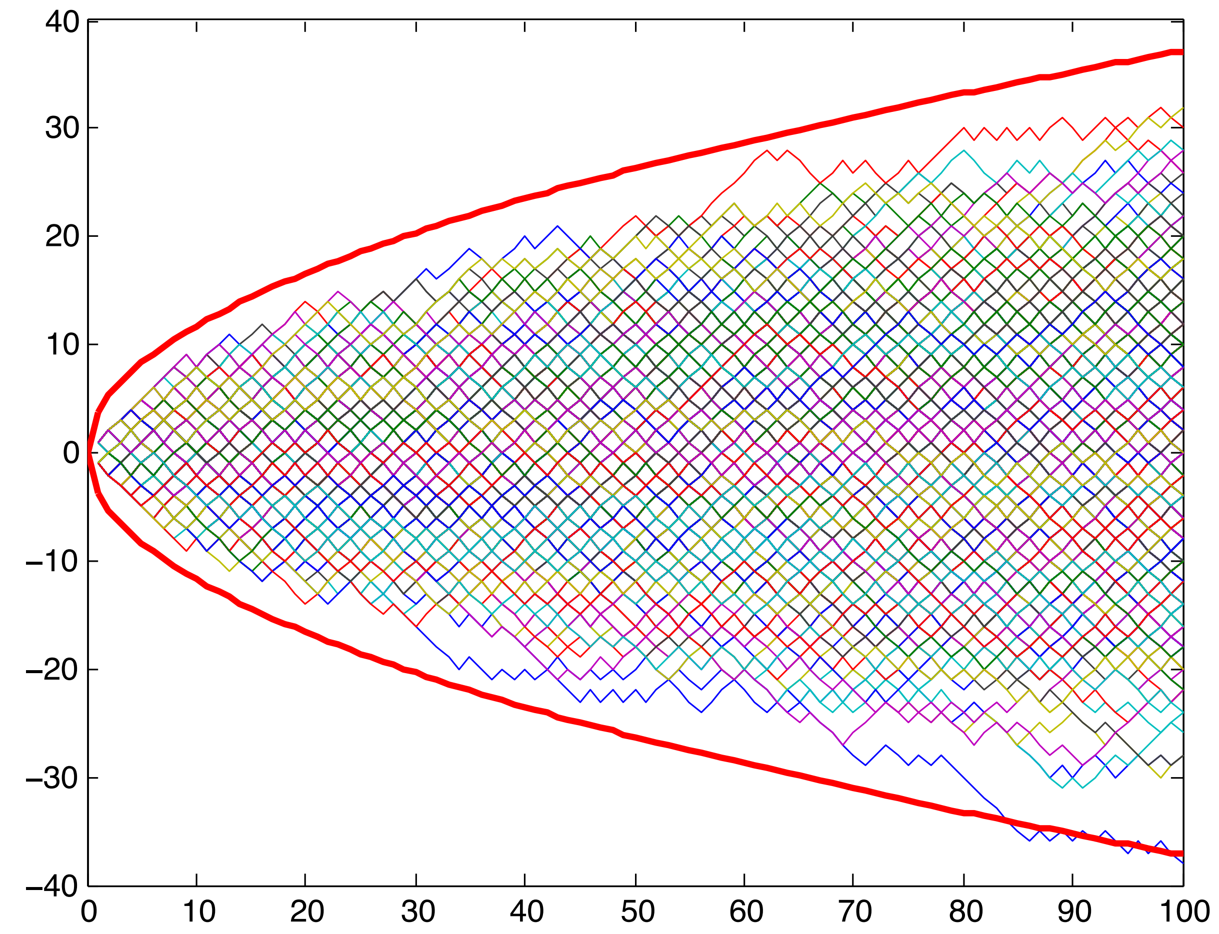
Equivalently $\forall i, R_i^T \le \sqrt{2T \ln N} + \ln N; \quad \lim_{T \to \infty} \frac{\sqrt{2T \ln N} + \ln N}{T} = 0$

Can we do better?

Lower bound

Each instantanous loss l_i^t is chosen IID -1/+1 with prob $\frac{1}{2}, \frac{1}{2}$

Cumulative loss defines a random walk.



Optimal weighting is always uniform.

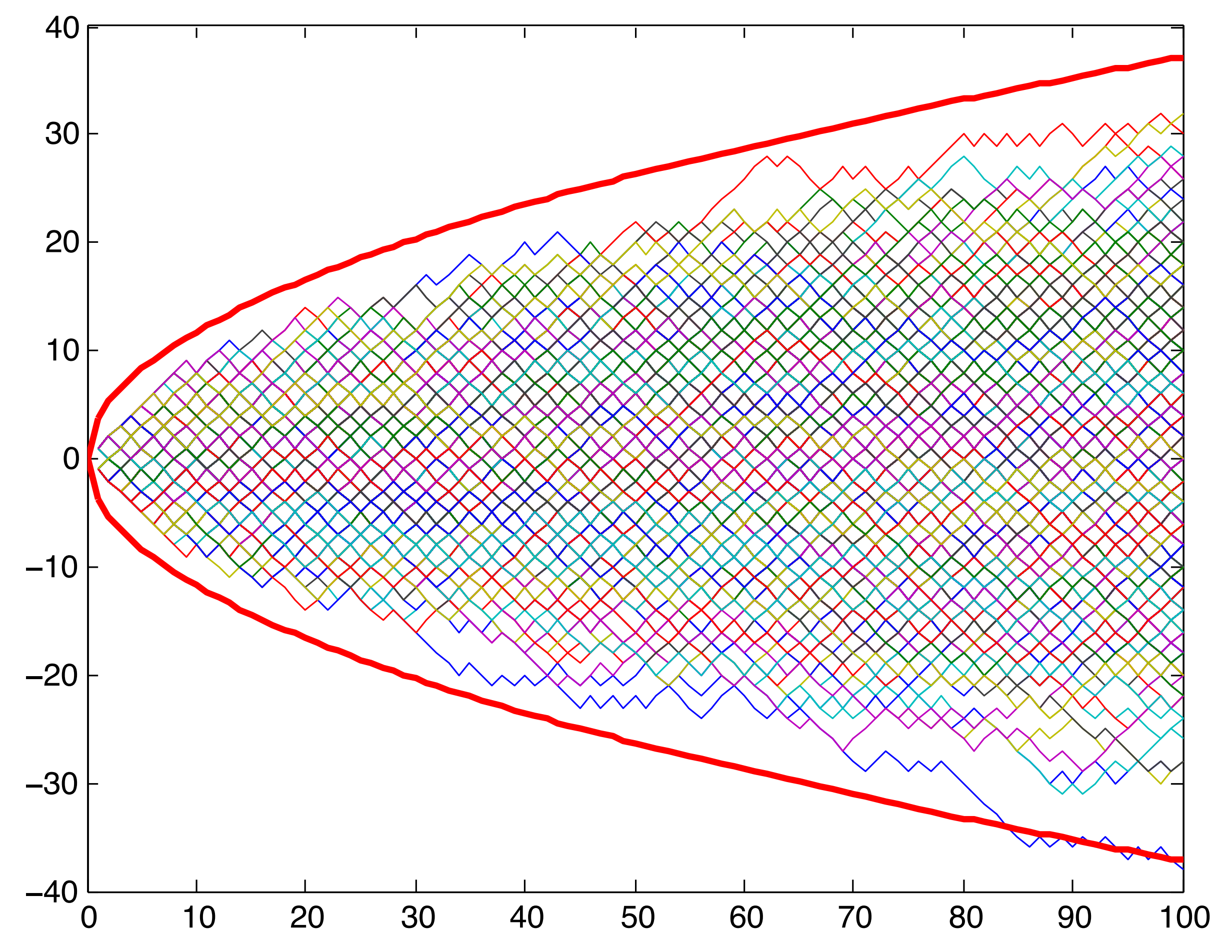
Optimal cumulative loss is always zero.

Link forward to BW

Lower bound

Each instantanous loss l_i^t is chosen IID -1/+1 with prob $\frac{1}{2}, \frac{1}{2}$

Cumulative loss defines a random walk.



Optimal weighting is always uniform.

Optimal cumulative loss is always zero.

With high probability one of the N actions

has cumulative loss smaller than $-\sqrt{2T \ln N}$

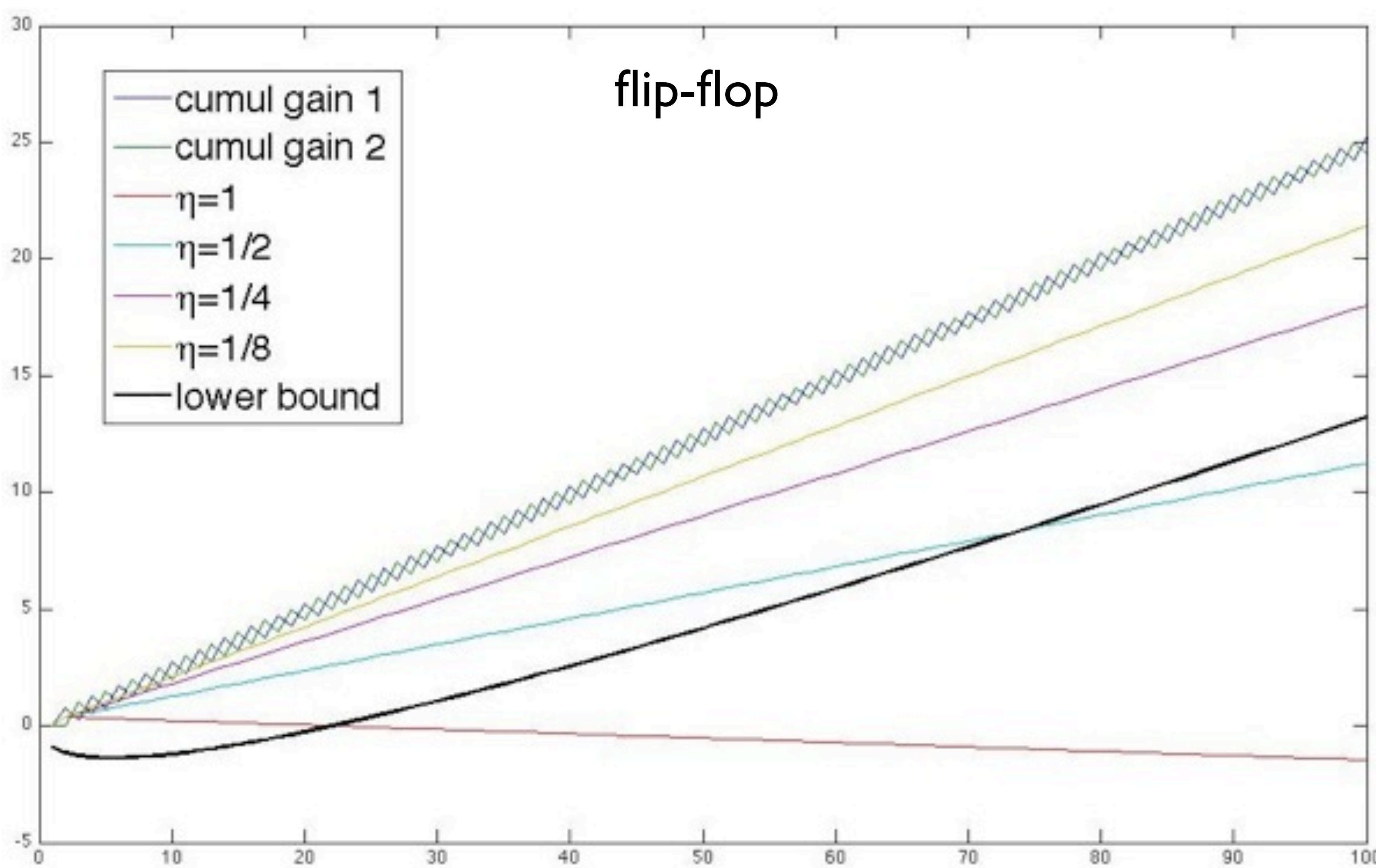
Link forward to BW

The problem with Hedge

If we set $\eta = \sqrt{\dfrac{2\ln N}{T}}$

Then we guarantee $\mathrm{L}_A^T \le \min_i L_i^T + \sqrt{2T\ln N} + \ln N$

Different setting of η for different T



N is not a real parameter

- If N actions consist of M groups, where in each group the behavior is identical, we want the bound to depend on M, not on N.
- If there uncountably many actions, we want a bound that depends on the fraction of actions that perform well.
- We want an algorithm with the optimal performance guarantee uniformly for N and for T.

ε-quantile instead of N

Instead of regret relative to best action,
compare performance to the best ε-quantile
i.e. L_ε s.t. for ε fraction of the actions $L_\theta < L_\varepsilon$

For Hedge we get:

$$W^t = \int_{[0,1]} e^{-L_\theta^t}\, dw(\theta) \geq \int_{\theta: L_\theta^t \leq L_\varepsilon} e^{-L_\theta^t}\, dw(\theta) \geq w\left(\theta : L_\theta^t \leq L_\varepsilon\right) e^{-L_\varepsilon}$$

If we set $\eta = \sqrt{\frac{-2\ln\varepsilon}{T}}$

Then we guarantee $\mathrm{L}_A^T \leq L_\varepsilon + \sqrt{-2T\ln\varepsilon} - \ln\varepsilon$

But we don't know either ε or T a-priori,
so we don't know how to set η

The NormalHedge potential

$$\text{Potential: } \psi(r,c) = \begin{cases} \exp\left(\dfrac{r^2}{2c}\right) & \text{if } r \geq 0 \\ 1 & \text{if } r \leq 0 \end{cases}$$

$$\text{Weight: } w(r,c) = \frac{\partial}{\partial r}\psi(r,c) = \begin{cases} \dfrac{r}{c}\exp\left(\dfrac{r^2}{2c}\right) & \text{if } r \geq 0 \\ 0 & \text{if } r \leq 0 \end{cases}$$

NormalHedge algorithm

for t=0,1,2,...

if $\forall i, R_i^t \le 0$ then $w_i^t = 1/N$

else

set $c(t)$ so that $\frac{1}{N}\sum_{i=1}^{N}\psi\left(R_i^t, c(t)\right) = e$

$w_i^t = w\left(R_i^t, c(t)\right)$

Incur instantanous losses: $\left\langle l_1^t, l_2^t, \ldots, l_N^t \right\rangle$

Algorithm loss: $l_A^t = \frac{\sum_{i=1}^{N} w_i^t l_i^t}{\sum_{i=1}^{N} w_i^t}$

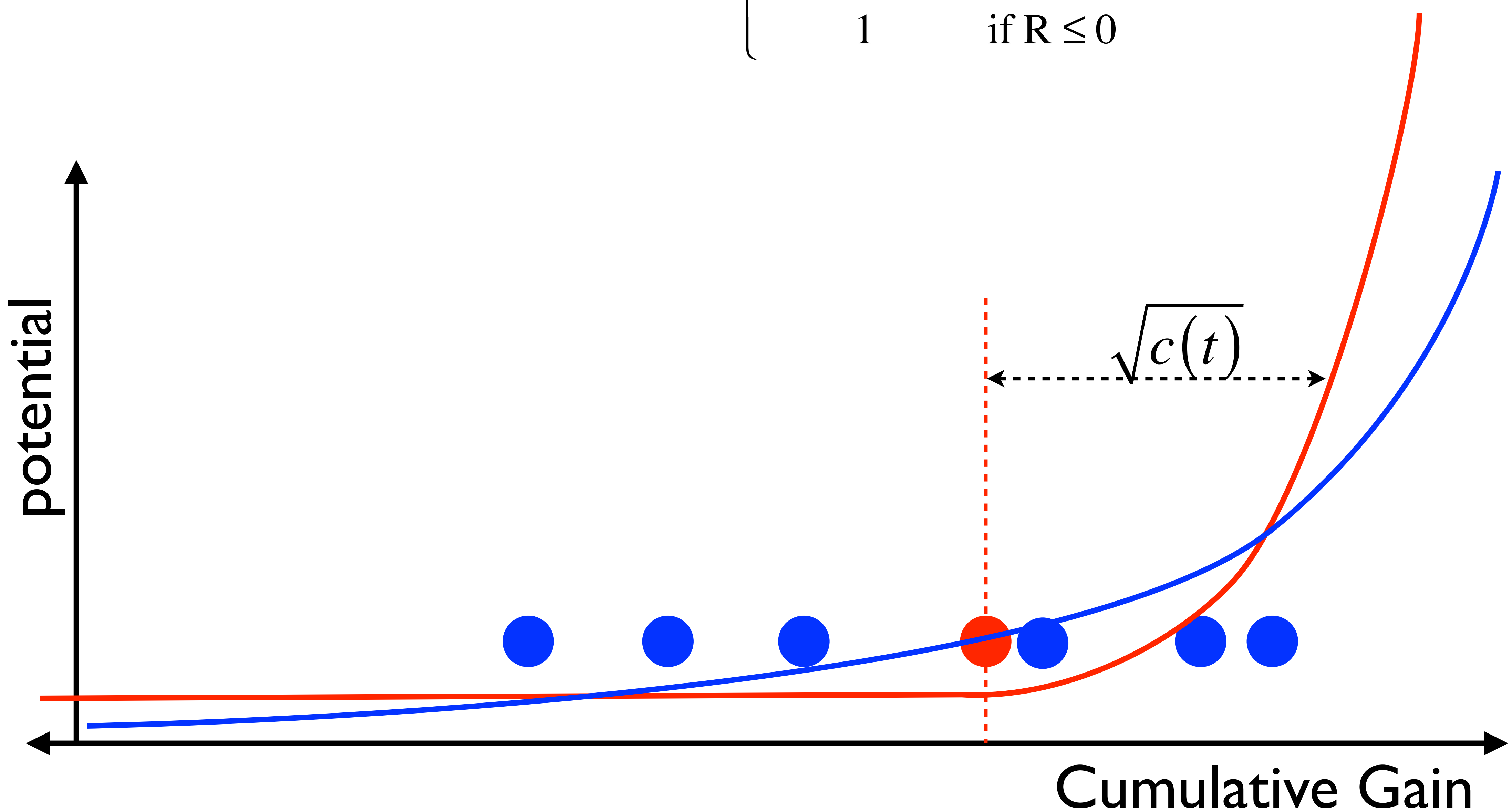
Update regrets: $R_i^{t+1} = R_i^t + l_A^t - l_i^t$

Illustrative Example

- Expert
- Algorithm

$\exp(\eta G)$

$$\begin{cases} \exp\left(\dfrac{R^2}{2c}\right) & \text{if R} \geq 0 \\ 1 & \text{if R} \leq 0 \end{cases}$$



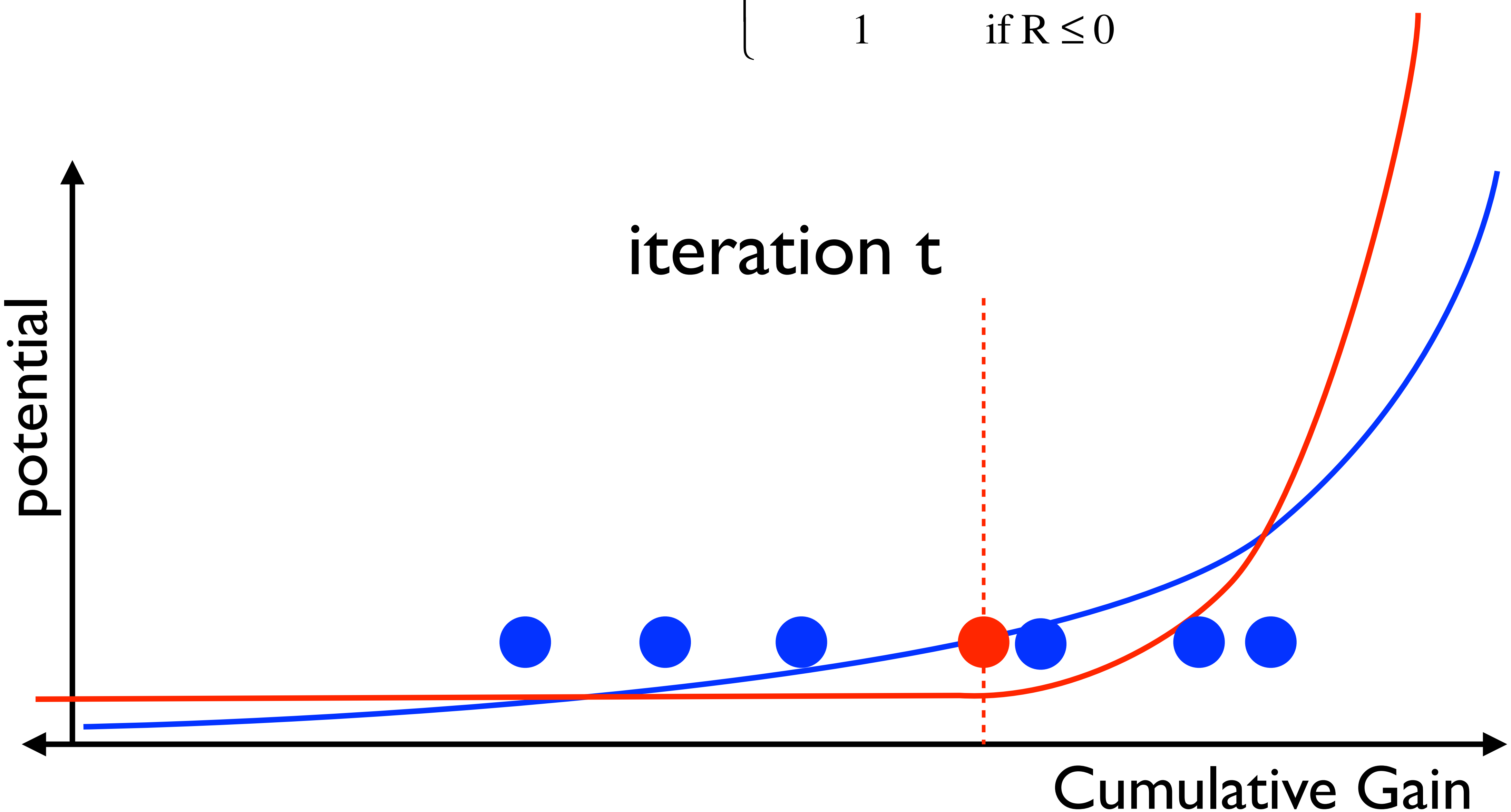
Illustrative Example

Expert

Algorithm

$$\exp(\eta G)$$

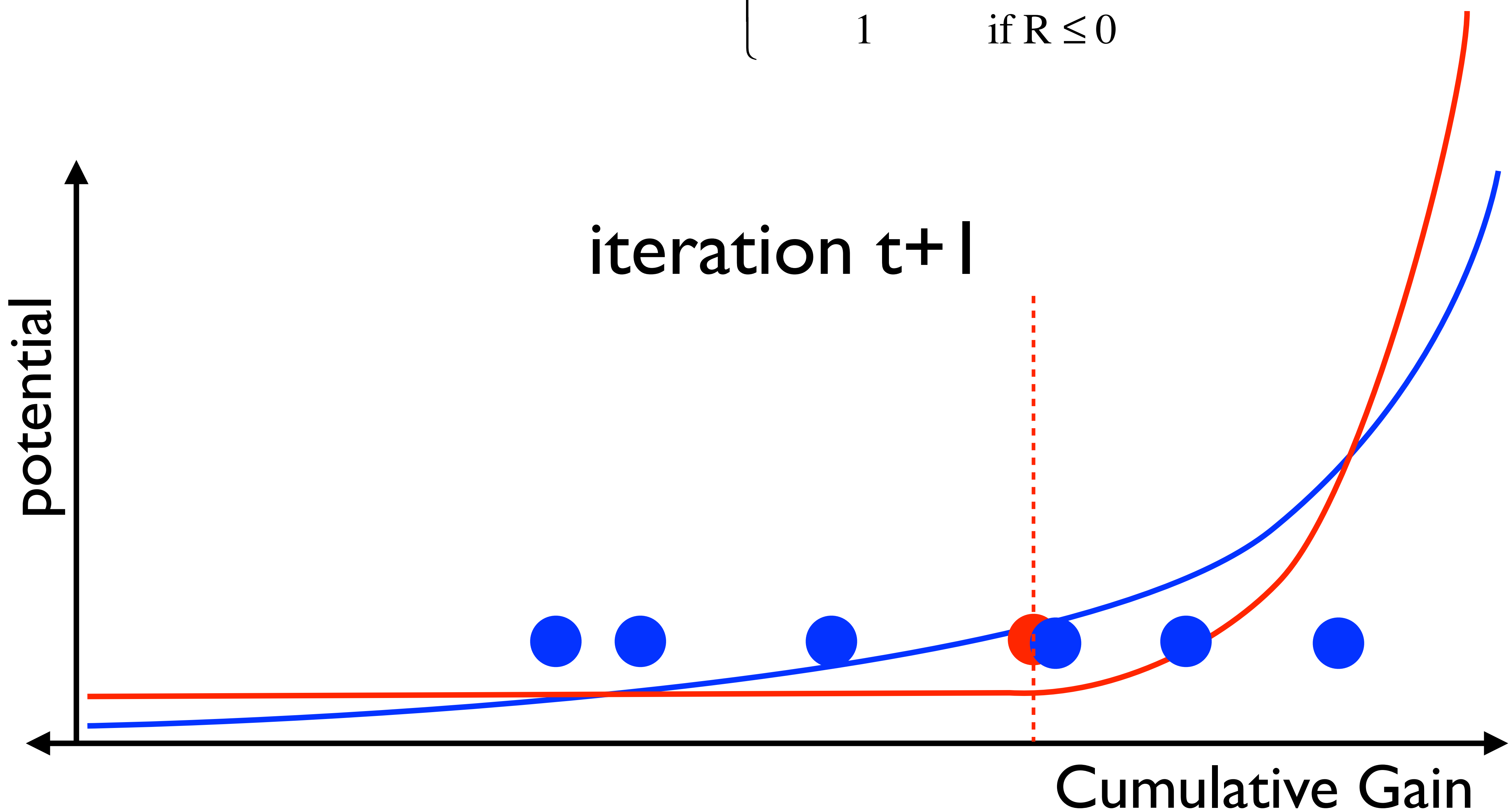
$$\begin{cases} \exp\left(\dfrac{R^2}{2c}\right) & \text{if R} \geq 0 \\ 1 & \text{if R} \leq 0 \end{cases}$$



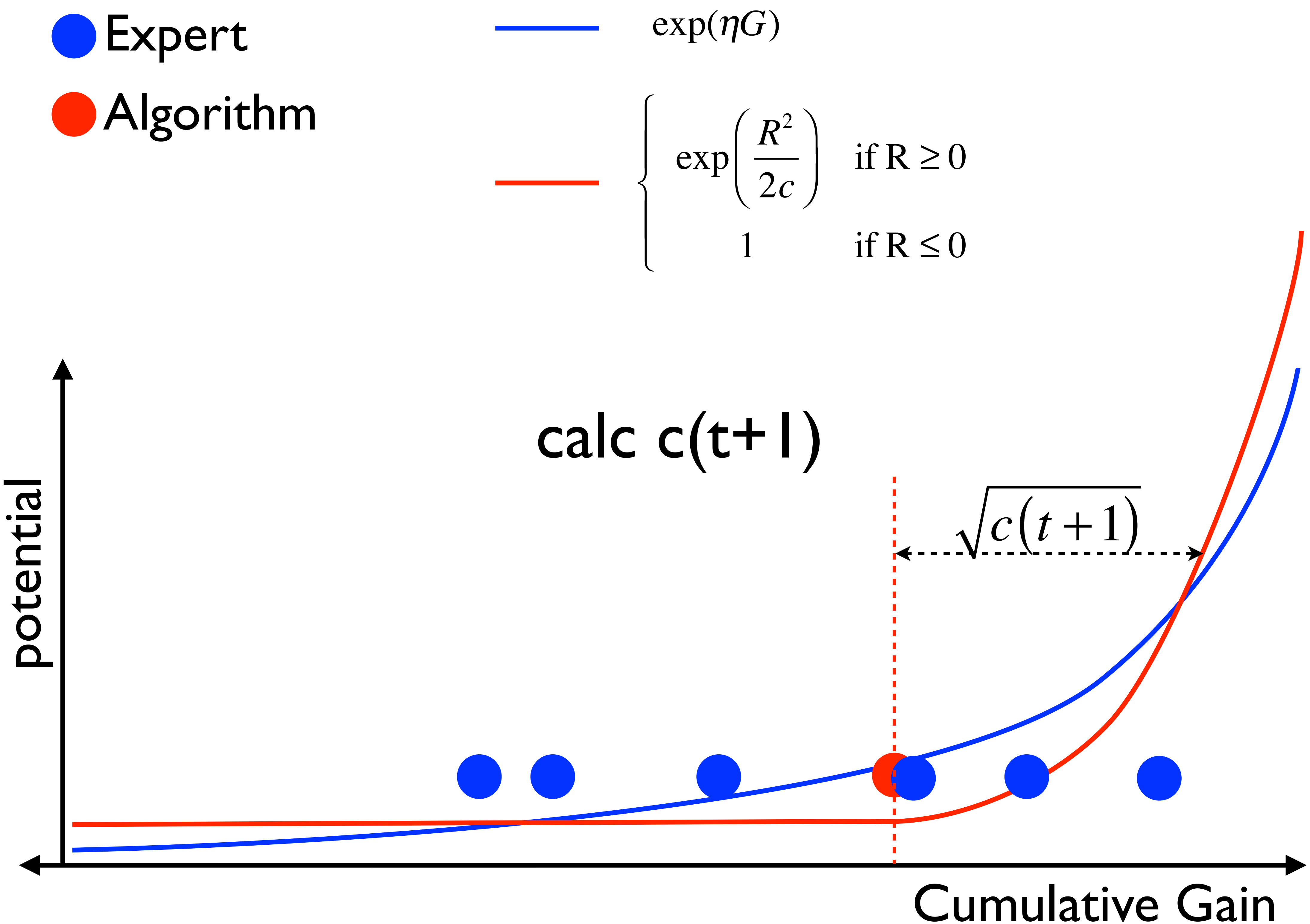
Illustrative Example

Expert $\exp(\eta G)$

Algorithm $\begin{cases} \exp\left(\dfrac{R^2}{2c}\right) & \text{if R} \geq 0 \\ 1 & \text{if R} \leq 0 \end{cases}$



Illustrative Example



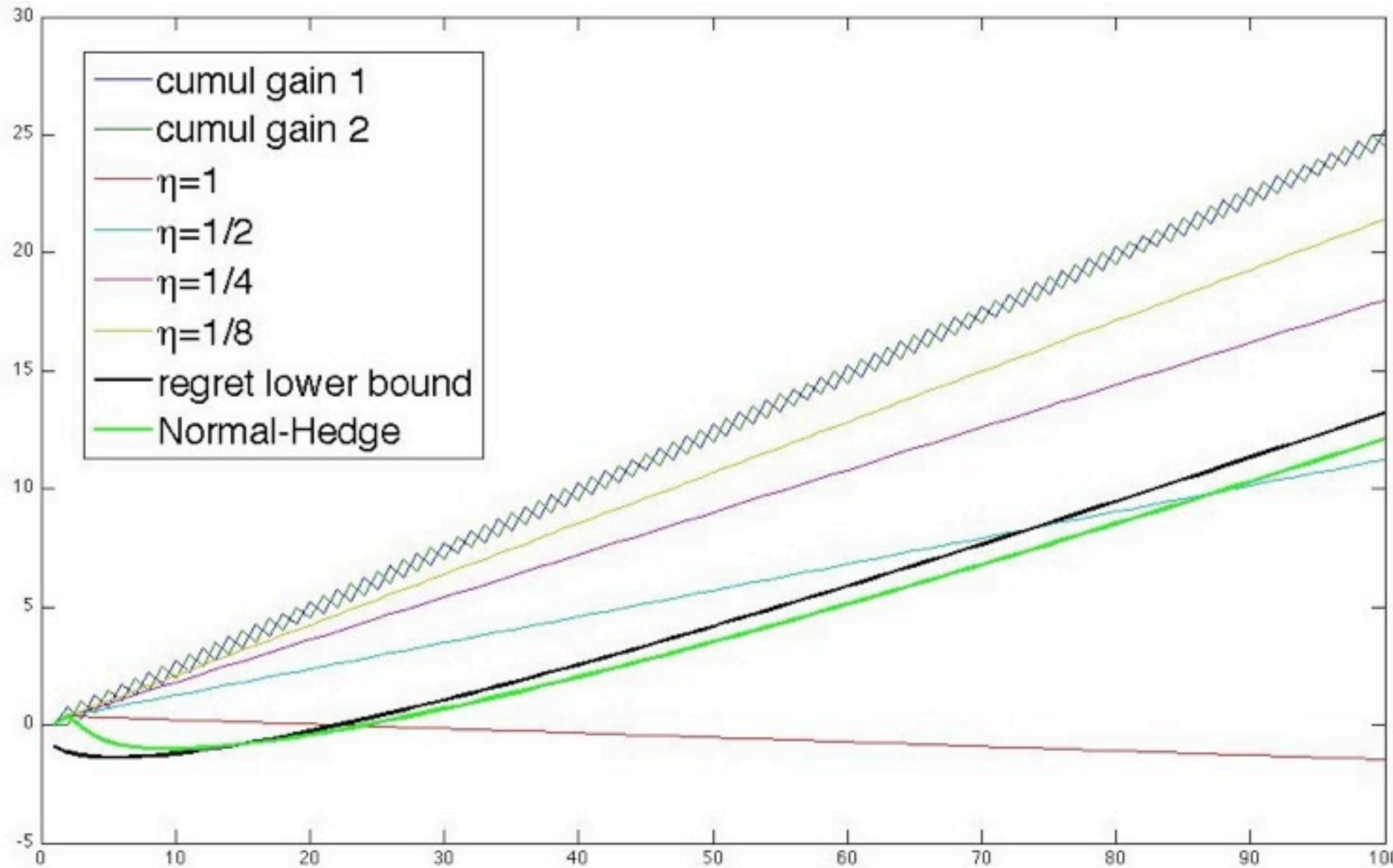
Normal-Hedge Performance bound

[Chaudhuri, Freund & Hsu 2009]

The regret of NormalHedge is upper bounded by

$$O\left(\sqrt{T \ln N + \ln^3 N}\right)$$

Performance on flip-flop



Combining experts, the binary prediction case

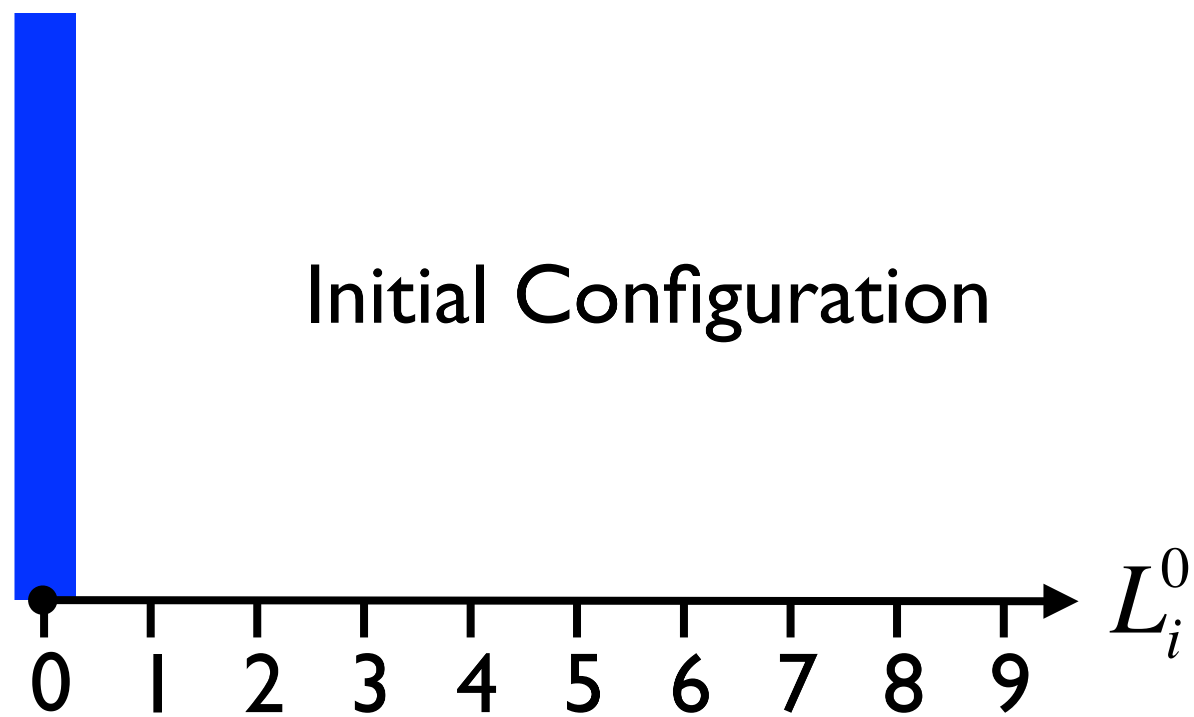
- Goal is to predict a binary sequence, making as few mistakes as possible.
- There are N experts.
- All predictions are binary and deterministic.
- A-priori knowledge: there is an expert that never makes more than k mistakes.
- k=0 corresponds to the halving algorithm.

Combining experts as a drifting game

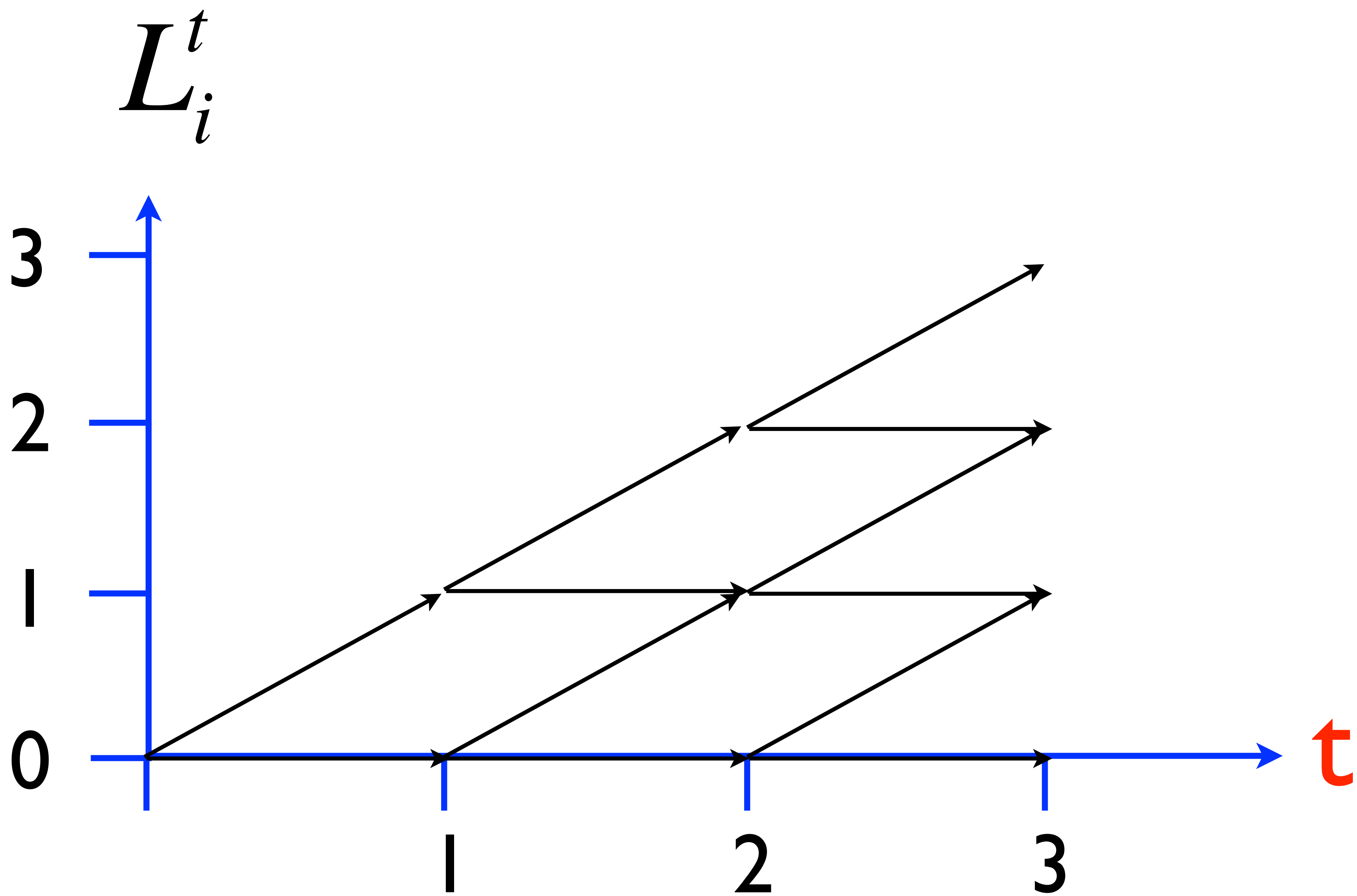
[Cesa-Bianchi, Freund, Helmbold, Warmuth 96]

Binary instantanous loss $l_i^t, l_A^t \in \{0,1\}$

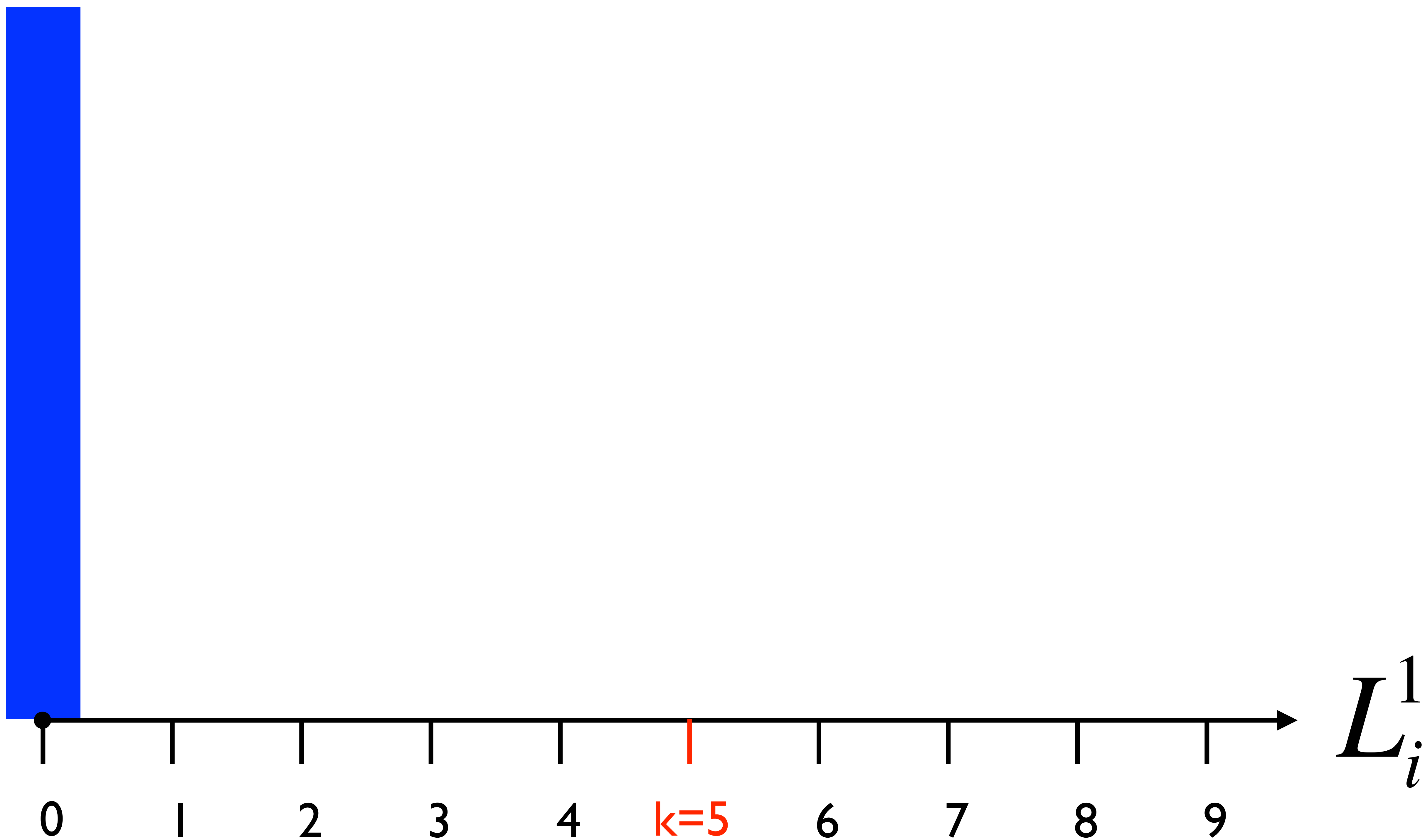
Bin s contains all experts for which $L_i^t = s$

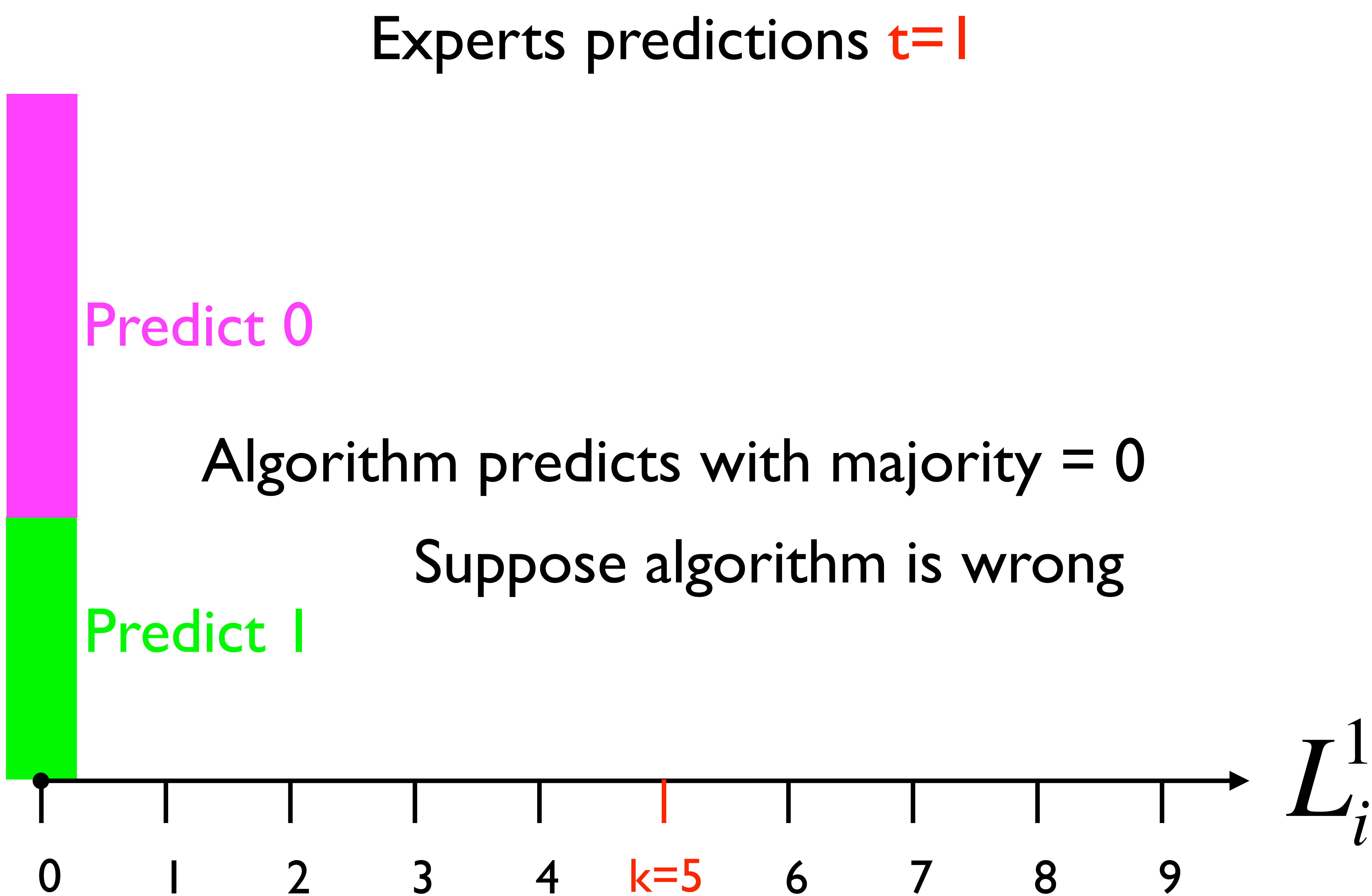


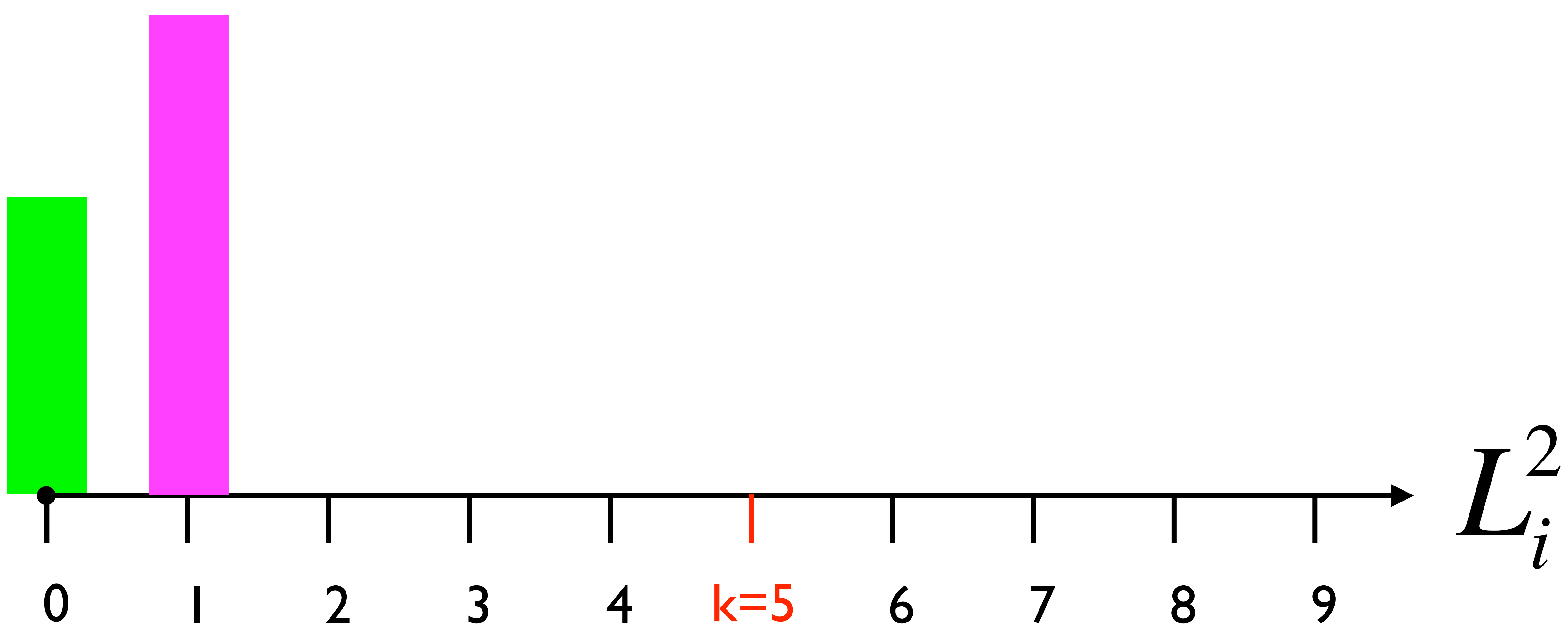
The game lattice



Initial configuration t=1

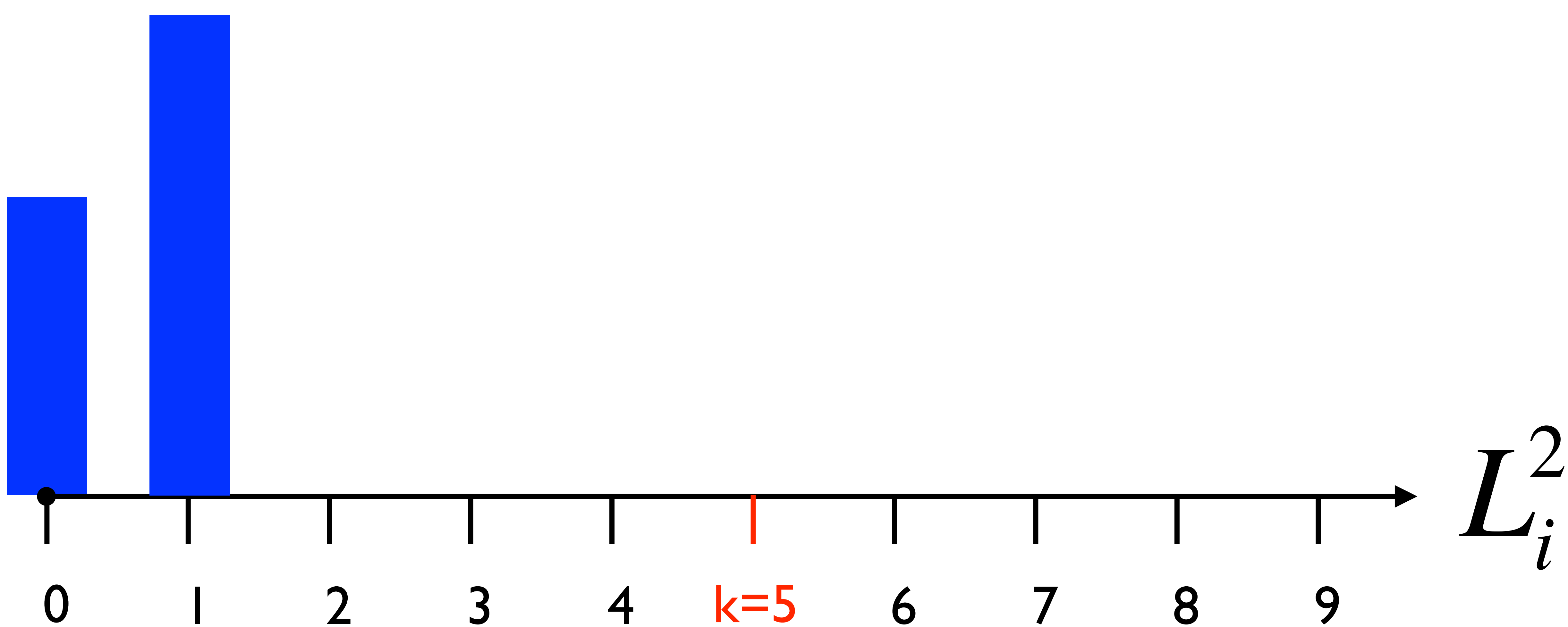



Experts predictions t=1
Predict 0
Algorithm predicts with majority = 0
Suppose algorithm is wrong
Predict 1
L_i^1
0
1
2
3
4
k=5
6
7
8
9

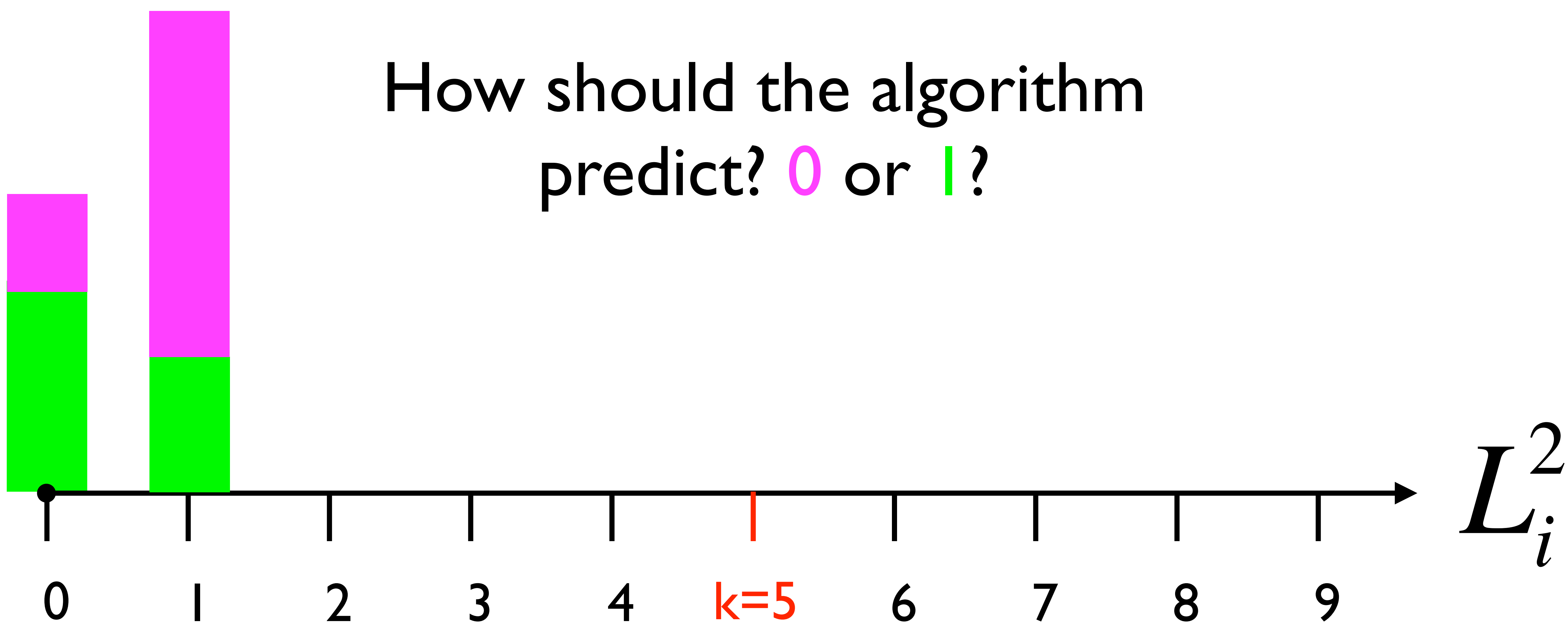


L_i^2
0
1
2
3
4
k=5
6
7
8
9

configuration at t=2



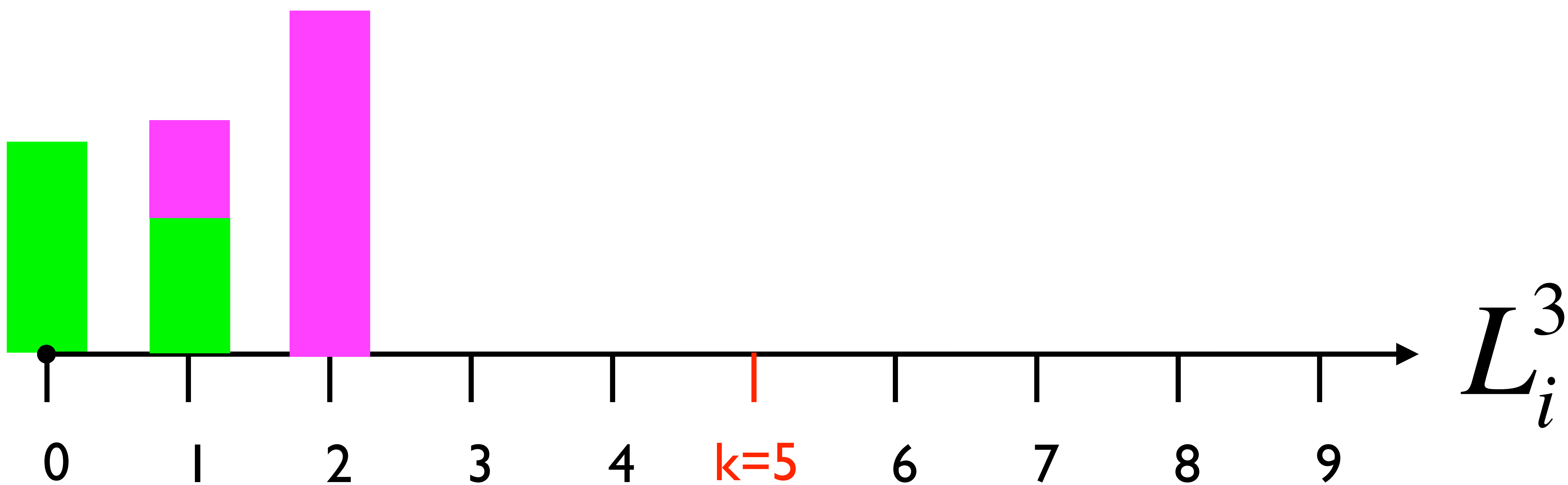
Experts predictions t=2



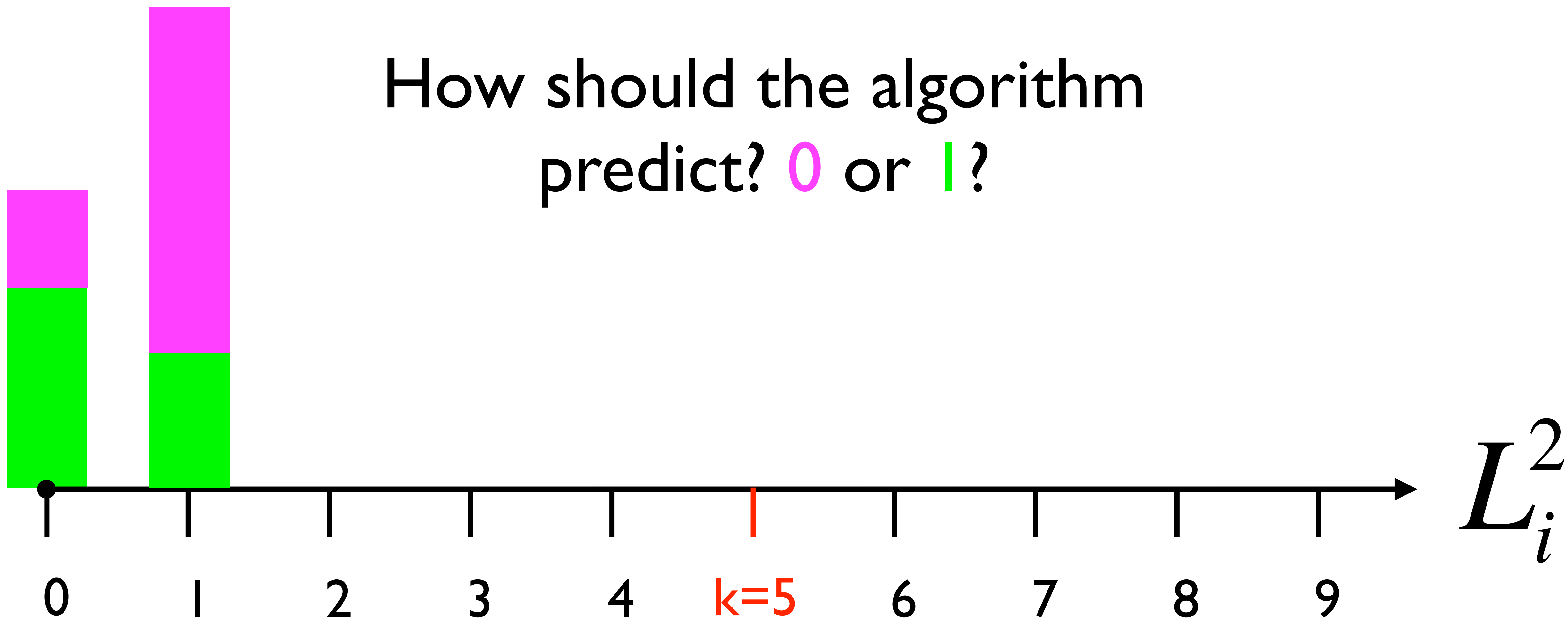
How should the algorithm predict? 0 or 1?

configuration t=3

Prediction is 0 and outcome is 1

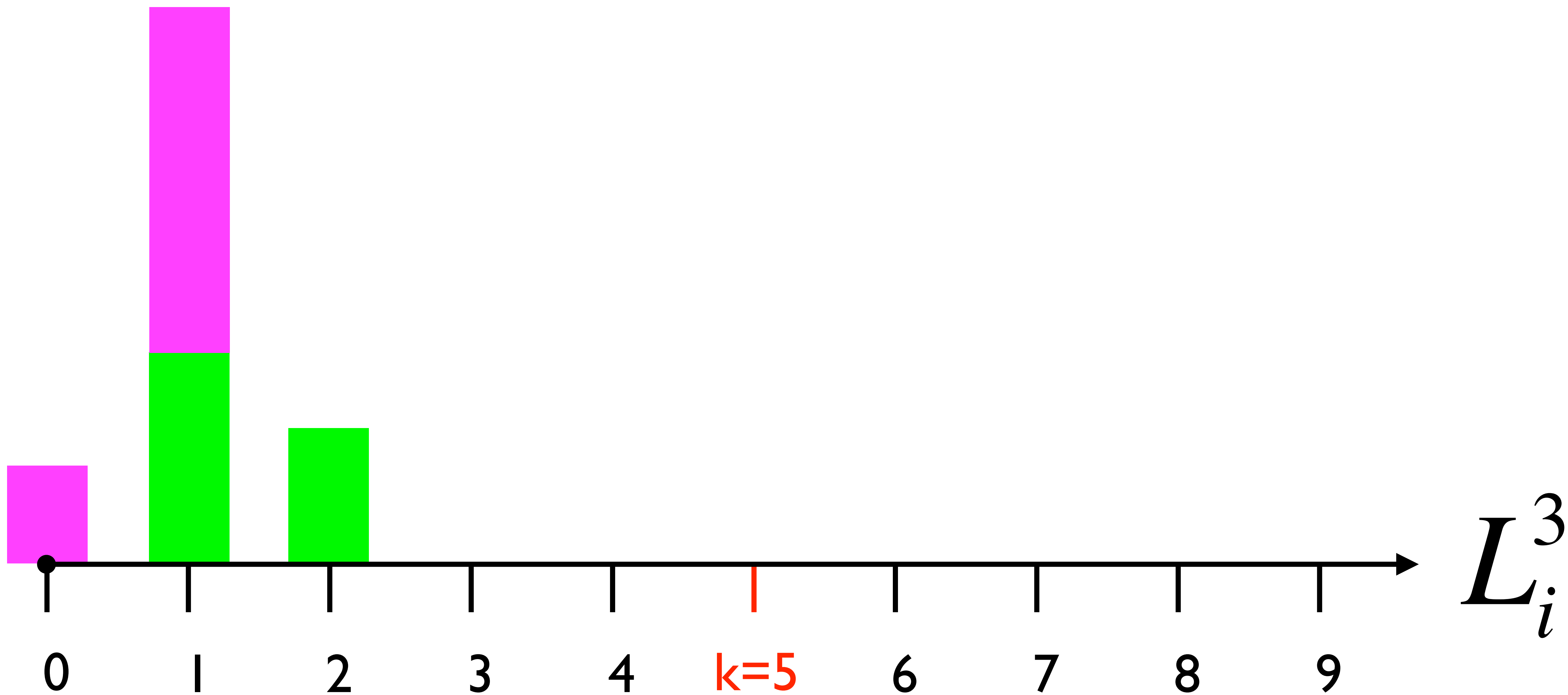


Experts predictions t=2



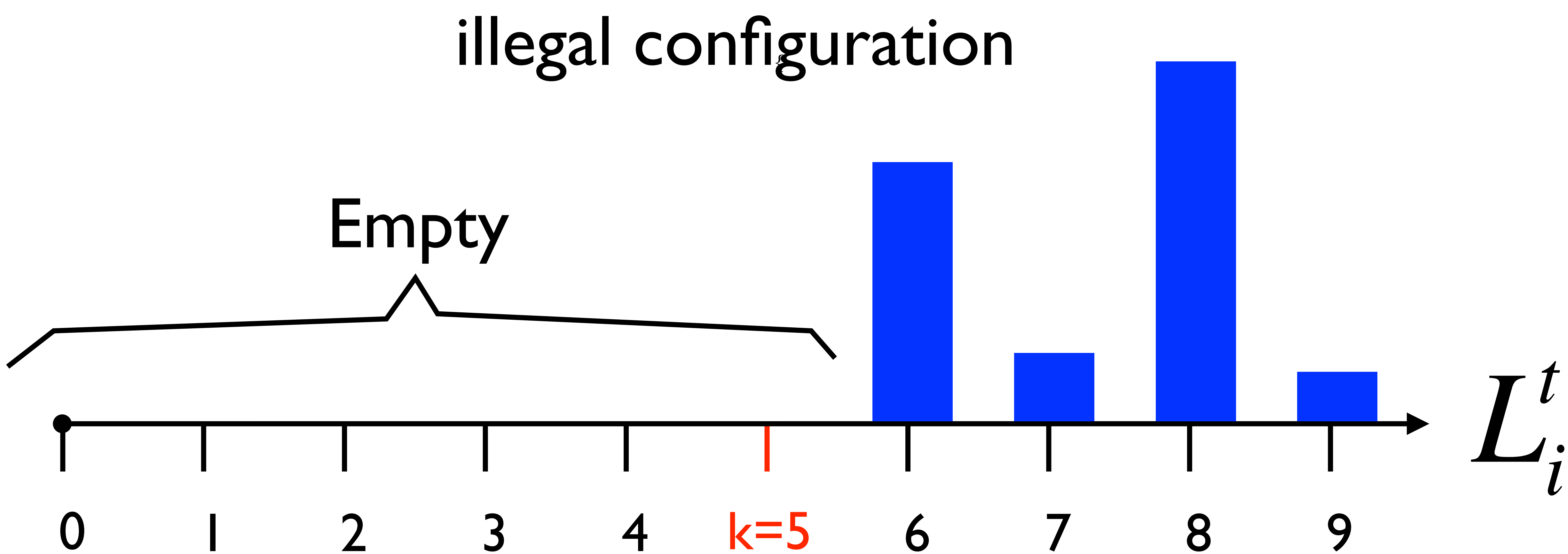
configuration t=3

Prediction is 1 and outcome is 0



If an error will lead to this configuration
then an error is not possible
$\Rightarrow$ this is a safe prediction

Algorithm's goal is to get to an illegal configuration
with the smallest number of mistakes.



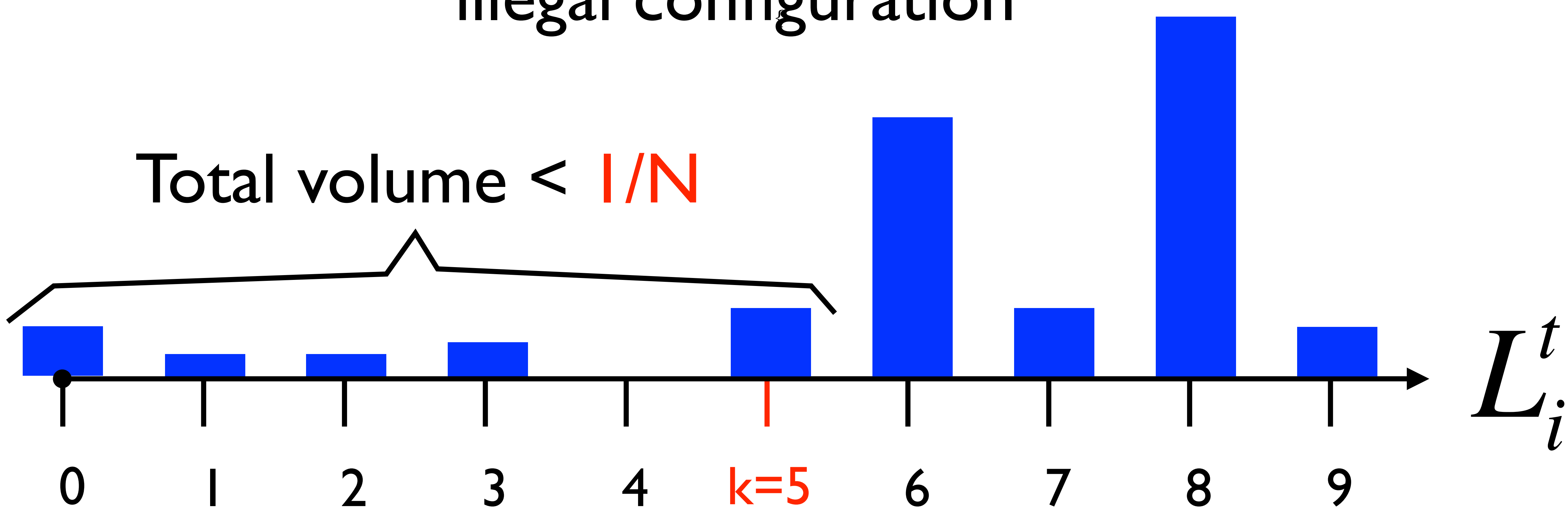
Helping the adversary.

- Assume that the set of experts is continuous, arbitrarily divisible.
- a-priori knowledge: $1/N$ fraction of the expert "mass" have cumulative loss at most k
- Find algorithm with the tightest uniform upper bound on the cumulative loss.

If an error will lead to this configuration
then an error is not possible
⇒ this is a safe prediction

Algorithm’s goal is to get to an illegal configuration with the smallest number of mistakes.

illegal configuration



An optimal adversarial strategy

- Split each bin to two equal parts. Algorithm's prediction is always incorrect.
- Equivalently: predictions of each expert are IID 0,1 with probabilities 1/2,1/2

An optimal adversarial strategy

- Split each bin to two equal parts. Algorithm's prediction is always incorrect.
- Equivalently: predictions of each expert are IID 0,1 with probabilities 1/2,1/2
- Same adversarial strategy was used to prove general lower bound on BLG

Link to lower bound

Optimal prediction strategy

- Assume that adversary will play optimally from the next iteration until the end of the game.
- Choose as the next configuration the one that would end the game faster.
- Relevant only when adversary plays sub-optimally, when adversary plays optimally the two next configurations are identical.

Potentials

- potential for bin i = the fraction of the experts in the bin that will have <k mistakes in r iterations.

$$\psi(i,r) = \mathrm{Binom}(k-i,r);\ \mathrm{Binom}(l,m) = \frac{1}{2^m}\sum_{j=0}^{l}\binom{m}{j}$$

- f(i) = the fraction of the experts currently in bin i

- potential for a configuration = weighted sum of bin potentials.

$$\Psi(\text{configuration}) = \sum_{j=1}^{k} f(i)\psi(i,r)$$

properties of the potential

$$\psi(i,r) = \frac{\psi(i,r-1)+\psi(i+1,r-1)}{2}$$

End of game

$$\psi(i,0) = \begin{cases} 1 & i \le k \\ 0 & i > k \end{cases}$$

Illegal configuration:

$$\Psi(\text{configuration}) = \sum_{j=1}^{k} f(i)\psi(i,0) < \frac{1}{N}$$

Beginning of game

Number of errors if adversary always plays optimally $r-1$, where r is the smallest integer for which

$$\psi(0,r) < \frac{1}{N}$$

BW Prediction algorithm

- Initialization: set r to be the number of errors against optimal adversary.
- Given expert predictions: choose prediction that will result (assuming error) in a lower-potential configuration.
- Decrease r if possible.

Main properties

- If algorithm is followed, the potential of the configuration never increases - is always $\leq$ 1/N
- Algorithm is min/max optimal.
  - Removing assumption that expert set is divisible min/max optimality holds if $N > 2^{2^k}$
  - Based on relation to Ulam's game with k lies [Spencer 92]

Alternative Representation

The difference between the two configurations can be represented as a weighted sum

$$\Psi(\text{configuration } 1) - \Psi(\text{configuration } 0) = \sum_{j=0}^{k} f(i)w(i,r)$$

$$w(i,r) = \psi(i+1,r-1) - \psi(i,r-1) = \frac{1}{2^{r-1}}\binom{r-1}{k-i}$$

The optimal prediction is according to the the sign of this weighted sum.

The BW algorithm

- Better error bound than exponential weights.
- A-priori assumption that one of the experts has loss at most k, we want a bound on the regret without any a priori assumptions.
- Instantaneous loss is restricted to $\{0,1\}$, we want it to be any number in $[-1,+1]$.

Design of NormalHedge

- BW: potential function depends on loss and number of remaining mistakes
- Normal-Hedge: Potential function based on regret and variance of the positive regrets

What next?

- I came up with the NormalHedge algorithm by considering the continuous time limit.
- The discrete-time proof is very technical and gives little insight.
- In the continuous time limit, the analysis is simple and insightful and the bound is much tighter.