



UvA-DARE (Digital Academic Repository)

Combining strategies efficiently: high-quality decisions from conflicting advice

Koolen, W.M.

Publication date
2011

[Link to publication](#)

Citation for published version (APA):

Koolen, W. M. (2011). *Combining strategies efficiently: high-quality decisions from conflicting advice*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

	y	n
y	\mathcal{G}_y	$\mathcal{G}_n + 1$
n	$\mathcal{G}_y + 1$	\mathcal{G}_n

2.1 Introduction

The simplest decision problem is the prediction of a binary outcome. For example, we would like to predict whether it is going to rain tomorrow or not, whether a stock's price will be higher tomorrow or lower, etc. Predicting a binary outcome is an important and common task, and it is also a fundamental decision problem because it arises as a special case of many more sophisticated decision problems. We already encountered this prediction problem in Section 1.5.1.

In this chapter we study predicting a binary outcome from the online learning viewpoint, with the goal of understanding both this fundamental decision problem and its solutions. While we present several new results, the chapter also serves an introductory purpose: this chapter requires no prior knowledge about online learning, and illustrates by example the concepts used in the subsequent four chapters. We deliberately keep the setup as simple as possible. That is, we adopt the typical loss function for this scenario, the 0/1 loss, which simply counts the number of mistakes.

We formulate online prediction with experts as a game, and use elementary game theory to solve it. The advantage of this approach is twofold. First, we get the *minimax regret*, a number that expresses the complexity of the binary prediction problem, and gives us a lower bound on the regret of any prediction algorithm. And second, we get the *minimax strategy*, a prediction algorithm that achieves that lower bound, and hence provides the optimal regret guarantee. This game-theoretic approach to online learning was generally believed to be intractable [27]. But two recent breakthrough papers found efficient minimax strategies for prediction [4] and decision-making [6]. These algorithms compete with *black-box* experts, whose predictions and losses are determined completely adversarially.

In this chapter we focus on the simpler *white-box experts* (see the taxonomy in Section 1.6), for which we can compute all future predictions. We first consider the simplest, constant experts. Unexpectedly, by simplifying the prediction problem to its core, we obtain several new results, including a new, optimal, efficient algorithm for the more complex switching experts and for other elaborate classes of white-box experts. We also obtain new insights in existing algorithms by observing how exactly they coincide for our simple setup.

Experts We abstract away the meaning of the outcome, and call the two possible values y and n . We use the two simplest *constant experts*

yyyyyyyyyyyyyyyyyy . . . and nnnnnnnnnnnnnnnnn . . .

That is, one expert always predicts y and the other always predicts n . Apart from simplicity, another motivation for using these two experts is the following. If we assume that the outcomes are generated by flipping a biased coin, say loaded towards y , then the expected $0/1$ loss is minimised by *always* predicting y , and vice versa for n . Hence the entire biased coin model is predicted optimally by these two experts.

Note however that the stochastic assumption made in the above argument is merely used as a sanity check to argue that these experts are reasonable, and that we never make any such unverifiable assumptions about the outcome-generating process in any of our prediction problems.

The goal now is to suffer small regret w.r.t. the best expert. In this case, competing with the best constant expert means learning the global trend, i.e. the outcome most frequent in the final sequence. Although seemingly simple, the prediction problems with these two constant experts exhibit many interesting general phenomena, while clearly illustrating the underlying difficulties.

We subsequently consider the more sophisticated *switching experts* that may switch between outcomes, predicting for example

yyyyynnnnnnyyyy . . .

Whereas competing with the best constant expert allows us to learn the global trend of the sequence of outcomes, competing with switching experts allows us to track local trends. This is useful in applications where we expect the outcome-generating process to sometimes change its characteristics, for example because it can be in several states. Again, the goal is to suffer small regret w.r.t. the best switching expert.

Adversary In each trial of the binary prediction problem, we have uncertainty about the actual outcome. We model this uncertainty by placing the generation of the actual outcome in the hands of a second player, which we call Adversary. The sole purpose of Adversary is to maximally frustrate our learning attempt. Thus we are trying to

minimise our regret while Adversary is trying to maximise it. Our goal is to find a strategy that guarantees small regret against the malevolent Adversary. By preparing for the worst case, we obtain a strategy with regret guarantees that *always* hold. In particular, the validity of our guarantees does not depend on any assumptions about the underlying outcome-generating process.

Goal & Outline In this chapter we model the prediction problem as a mathematical game, stressing the strategic considerations underlying its analysis. We first review elementary strategic game theory in Section 2.2, and then use it to solve predicting a single binary outcome in Section 2.3. We then review the theory of repeated games in Section 2.4, and use it to solve predicting a sequence of binary outcomes in Section 2.5. We apply the solution to related problems in Section 2.6. We then consider the special case that the best expert has small cumulative loss in Section 2.7, and obtain regret bounds that are parametrised accordingly. We demonstrate the generality of the methodology in Section 2.8 and apply it to the set of switching experts in Section 2.9. Section 2.11 concludes by placing these results into context.

2.2 2×2 Strategic Games

In this chapter, we model several prediction problems as mathematical games, and this section introduces the game theory necessary to model a single trial. The game format appropriate for our problems is that of strategic games, in which the players choose their action in parallel. Contemporaneous moves model both our uncertainty about the actual outcome, and our ability to randomise our prediction in a way that Adversary cannot foretell.

In general, a two-player zero-sum 2×2 strategic game is represented by a cost matrix:

		Adversary	
		y	n
We	y	A	B
	n	C	D

We identify with the row player, and call the column player Adversary.

Both players independently choose an action, either y or n . We then incur the cost A, B, C or D associated with the chosen pair of actions. It is our objective to minimise the cost, while Adversary tries to maximise it. In this section we compute our optimal strategy and its guarantee. Our optimal strategy is often a *mixed strategy*, which carefully randomises its action.

Saddle Point We identify a mixed strategy for either player with the probability it assigns to the action y . When we follow strategy $\sigma \in [0, 1]$, while Adversary follows strategy $\tau \in [0, 1]$, our expected cost is

$$V(\sigma, \tau) := \sigma\tau A + \sigma(1 - \tau)B + (1 - \sigma)\tau C + (1 - \sigma)(1 - \tau)D.$$

A *minimax* strategy $\hat{\sigma}$ attains $\min_{\sigma} \max_{\tau} V(\sigma, \tau)$ while a *maximin* strategy $\hat{\tau}$ attains $\max_{\tau} \min_{\sigma} V(\sigma, \tau)$. The celebrated Von Neumann minimax theorem (see [16]) states that a saddle point $\langle \hat{\sigma}, \hat{\tau} \rangle$ exists with

$$V(\hat{\sigma}, \hat{\tau}) = \min_{\sigma} \max_{\tau} V(\sigma, \tau) = \max_{\tau} \min_{\sigma} V(\sigma, \tau).$$

We call the above quantity the *game value* and denote it by \mathcal{V} . The game value \mathcal{V} equals both our *minimax cost* and Adversary's *maximin cost*. In words, our minimax strategy $\hat{\sigma}$ guarantees that our expected cost is at most \mathcal{V} , while Adversary's maximin strategy $\hat{\tau}$ guarantees that our expected cost is at least \mathcal{V} . Consequently, playing $\hat{\sigma}$ vs $\hat{\tau}$ results in \mathcal{V} .

Computing a Saddle Point For 2×2 games it is straightforward to compute the game value \mathcal{V} and construct a saddle point $\langle \hat{\sigma}, \hat{\tau} \rangle$, as shown in Table 2.1. Games satisfying any of the first four cases of Table 2.1 have a saddle point in *pure*, i.e. deterministic, strategies. The interesting case is the fifth, where randomisation is essential for both players. A tedious but straightforward case-by-case analysis shows that no pure saddle point exists if and only if

$$(A - B)(C - D) < 0 \quad \text{and} \quad (A - C)(B - D) < 0. \quad (2.1)$$

We now solve games without pure saddle points. The game-theoretic analyses in all later sections are built upon this workhorse lemma.

Table 2.1 Solution of 2×2 matrix games. The game value \mathcal{V} and saddle point $(\hat{\sigma}, \hat{\tau})$ are shown as a function of the costs A, B, C and D .

$\hat{\sigma}$	$\hat{\tau}$	\mathcal{V}	condition
1	1	A	$C \geq A \geq B$
1	0	B	$D \geq B \geq A$
0	1	C	$A \geq C \geq D$
0	0	D	$B \geq D \geq C$
$\frac{D-C}{Z}$	$\frac{D-B}{Z}$	$\frac{AD-BC}{Z}$	condition (2.1)

where $Z = A - B - C + D$

2.2.1. LEMMA. *If a game has no saddle point in pure strategies, then the game value \mathcal{V} and unique saddle point $(\hat{\sigma}, \hat{\tau})$ are given by*

$$\mathcal{V} = \frac{AD - BC}{Z}, \quad \hat{\sigma} = \frac{D - C}{Z} \quad \text{and} \quad \hat{\tau} = \frac{D - B}{Z}$$

where $Z = A - B - C + D$.

Proof. Say that we play action y with probability $\sigma \in [0, 1]$. We can then guarantee expected cost

$$\max_{\tau \in [0,1]} V(\sigma, \tau) = \max\{\sigma A + (1 - \sigma)C, \sigma B + (1 - \sigma)D\}.$$

The right hand is a maximum of two linear functions. We now use (2.1). Since $(A - C)(B - D) < 0$ one function is increasing in σ while the other is decreasing, so their maximum is minimised at their intersection. Moreover, this intersection occurs for $\sigma \in (0, 1)$ because $(A - B)(C - D) < 0$. Hence the unique $\hat{\sigma}$ is found by solving for σ in

$$\sigma A + (1 - \sigma)C = \sigma B + (1 - \sigma)D.$$

The computation of $\hat{\tau}$ is analogous, and simple algebra yields \mathcal{V} . \square

Equaliser Strategies The proof of the lemma in fact shows something more, namely that both $\hat{\sigma}$ and $\hat{\tau}$ are *equaliser strategies*, that is, $V(\hat{\sigma}, \tau) = V(\sigma, \hat{\tau}) = \mathcal{V}$ for all σ and τ . The equaliser strategy $\hat{\sigma}$ removes all power from Adversary by rendering all her strategies exactly equally costly for us.

Expectations We stress that all expectations in this chapter refer to strategic randomness deliberately introduced by the players to realise their guarantees. In particular, we never assume that outcomes are drawn from a true distribution.

With now apply this machinery to forecasting a binary outcome.

2.3 Predicting a Single Binary Outcome

We first model the single-trial problem as a game. Both we and Adversary choose a value in $\{y, n\}$. We call our value the *prediction* and Adversary's value the *actual outcome*. We make a mistake if our prediction does not equal the actual outcome. Recall that there are two experts, one that predicts y and one that predicts n . Since, in the single-trial case, the best expert makes no mistake, our regret equals our loss. Taking our regret as our cost results in the *one-shot regret game*:

$$\mathcal{G} := \begin{array}{cc} & \text{Adversary} \\ & \begin{array}{cc} y & n \end{array} \\ \text{We} & \begin{array}{cc} y & \begin{array}{|c|c|} \hline 0 & 1 \\ \hline \end{array} \\ n & \begin{array}{|c|c|} \hline 1 & 0 \\ \hline \end{array} \end{array} \end{array}$$

Both our pure strategies y and n guarantee that we make at most one mistake. This guarantee is trivial, since it is the maximal number of mistakes. We now show that randomisation improves our guarantee.

2.3.1. THEOREM. *The one-shot regret game \mathcal{G} has minimax regret \mathcal{V} and unique saddle point $\langle \hat{\sigma}, \hat{\tau} \rangle$, where*

$$\mathcal{V} = 1/2, \quad \hat{\sigma} = 1/2 \quad \text{and} \quad \hat{\tau} = 1/2.$$

Proof. By (2.1) \mathcal{G} has no saddle point in pure strategies, and hence the unique saddle point and game value are given by Lemma 2.2.1. \square

The strategy $\hat{\sigma}$, which predicts uniformly at random, is an equaliser strategy that guarantees that we make a mistake with probability exactly a half. That is, all actual outcomes are equally adversarial.

In a sense, predicting a single binary outcome with two experts is not very interesting because we have *too many* (2) experts relative to the number of outcomes (2): so many that the best expert always has loss

zero. The situation is very different if we predict a sequence of multiple binary outcomes in a row. We first introduce the necessary repeated-game theory and subsequently analyse the scenario with repetition.

2.4 Repeated Games

A repeated game is a matrix game that results in games [16]. Such games are defined recursively. In the elementary (zero-round) game $V \in \mathbb{R}$ we simply suffer cost V . Then if $\mathcal{H}, \mathcal{I}, \mathcal{J}$ and \mathcal{K} are games, so is the composite game

$$\begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|c|c|} \hline \mathcal{H} & \mathcal{I} \\ \hline \mathcal{J} & \mathcal{K} \\ \hline \end{array} \end{array} \end{array}$$

in which the two players independently choose an action, and jointly determine the subsequent game to be played. An example of a two-trial game is

$$\begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|c|c|} \hline \begin{array}{cc} y & n \\ \hline y & \begin{array}{|c|c|} \hline A & B \\ \hline C & D \\ \hline \end{array} & \begin{array}{cc} y & n \\ \hline y & \begin{array}{|c|c|} \hline E & F \\ \hline G & H \\ \hline \end{array} & \end{array} \\ \hline \end{array} \end{array} \cdot \end{array} \quad (2.2)$$

Say that, in the first trial, we play y and Adversary plays n . Together, these actions determine that the players subsequently face the constituent game

$$\begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|c|c|} \hline E & F \\ \hline G & H \\ \hline \end{array} \end{array} \cdot \end{array} \quad (2.3)$$

If, in the second trial, we play n and Adversary plays n , then the players subsequently face the elementary game

$$H, \quad (2.4)$$

in which we immediately incur the specified cost H .

A *history* of length t is an element of $\{y, n\}^t \times \{y, n\}^t$. A history $\langle p, x \rangle$ of length t records our prediction p_i and the actual outcome x_i for each played trial $1 \leq i \leq t$. The subgame of a game \mathcal{G} identified by the history $\langle p, x \rangle$ is denoted by $\mathcal{G}_{p,x}$. For example, let \mathcal{G} denote the game displayed as (2.2). The history $\langle y, n \rangle$ identifies the subgame $\mathcal{G}_{y,n}$ displayed as (2.3), while the history $\langle yn, nn \rangle$ identifies the 0-round subgame $\mathcal{G}_{yn,nn}$ displayed as (2.4). The *longer* the history, the *shorter* the subgame it identifies. The empty history $\langle \epsilon, \epsilon \rangle$ identifies the entire game itself: $\mathcal{G} = \mathcal{G}_{\epsilon,\epsilon}$ for any game \mathcal{G} .

As before, we are looking for the optimal strategies and for the game value, i.e. the best possible expected cost guarantee. Repeated games can be solved recursively by the procedure called *backwards induction*, also known as *dynamic programming* or *Zermelo's algorithm* [16].

2.4.1 Backwards Induction

Solving elementary games is trivial. To solve a composite game

$$\mathcal{G} = \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|cc|} \hline \mathcal{G}_{y,y} & \mathcal{G}_{y,n} \\ \hline \mathcal{G}_{n,y} & \mathcal{G}_{n,n} \\ \hline \end{array} \end{array},$$

first recursively solve its four constituent games to obtain their best achievable expected cost guarantees $\mathcal{V}_{y,y}$, $\mathcal{V}_{y,n}$, $\mathcal{V}_{n,y}$, and $\mathcal{V}_{n,n}$. Second, replace each constituent game by its value to obtain the single-trial game

$$\mathcal{G}' = \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|cc|} \hline \mathcal{V}_{y,y} & \mathcal{V}_{y,n} \\ \hline \mathcal{V}_{n,y} & \mathcal{V}_{n,n} \\ \hline \end{array} \end{array}.$$

The game \mathcal{G}' represents the game \mathcal{G} , assuming that we play all its subgames optimally. By solving the single-trial game \mathcal{G}' using Section 2.2, we therefore obtain the minimax strategy $\hat{\sigma}$, maximin strategy $\hat{\tau}$ and value \mathcal{V} of the composite game \mathcal{G} .

Backwards induction recursively solves all subgames of \mathcal{G} . For each subgame $\mathcal{G}_{p,x}$, we denote its game value by $\mathcal{V}_{p,x}$ and its saddle point by $\langle \hat{\sigma}_{p,x}, \hat{\tau}_{p,x} \rangle$.

2.5 Predicting a Sequence of Binary Outcomes

We now formalise the recurring prediction problem as a repeated game. We consider the case where we sequentially predict T binary outcomes, for some known fixed time horizon T . Recall that we have two experts, one that always predicts y and one that always predicts n , and that our goal is to minimise our regret w.r.t. the best expert.

Fix a sequence of predictions p and of actual outcomes x , both of length T . Our cumulative loss L and the cumulative loss of the best expert L^* are measured in mistakes, i.e.

$$L(p, x) := \sum_{t=1}^T \mathbf{1}_{p_t \neq x_t} \quad \text{and} \quad L^*(x) := \min \left\{ \sum_{t=1}^T \mathbf{1}_{y \neq x_t}, \sum_{t=1}^T \mathbf{1}_{n \neq x_t} \right\},$$

and our regret is thus

$$L(p, x) - L^*(x). \quad (2.5)$$

The regret is always evaluated at sequences of length exactly T , but it is convenient to allow the cumulative losses L and L^* to be evaluated at all p and x of equal length $t \leq T$, with in particular $L(\epsilon, \epsilon) = L^*(\epsilon) = 0$.

2.5.1 The Regret Game with Time Horizon T

We now formalise the sequential prediction problem with two experts as a repeated game, that we call the *regret game with time horizon T* . We first define its subgames $\mathcal{G}_{p,x}$ recursively. To each history $\langle p, x \rangle$ of length $t = T$ we assign the elementary game in which we incur the regret

$$\mathcal{G}_{p,x} := L(p, x) - L^*(x), \quad (2.6a)$$

while for each history $\langle p, x \rangle$ of length $t < T$ we simply play one trial

$$\mathcal{G}_{p,x} := \begin{array}{c} \begin{array}{cc} & \begin{array}{cc} y & n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|cc|} \hline \mathcal{G}_{py,xy} & \mathcal{G}_{py,xn} \\ \hline \mathcal{G}_{pn,xy} & \mathcal{G}_{pn,xn} \\ \hline \end{array} \end{array}. \quad (2.6b)$$

The *regret game with time horizon T* starts from the empty history $\langle \epsilon, \epsilon \rangle$

$$\mathcal{G} := \mathcal{G}_{\epsilon, \epsilon}. \quad (2.6c)$$

Predicting like the best expert amounts to solving the game \mathcal{G} . Our minimax strategy approximately predicts like the best expert, and our minimax regret quantifies the discrepancy. We solve the repeated game (2.6) by backwards induction. To simplify this task, we first use two properties of the specific cost function (2.6a), the regret, to simplify the representation of \mathcal{G} and its subgames.

2.5.2 The Regret Game (Simplified)

The regret (2.5) is a special cost function. First, our cumulative loss L is a sum that decomposes over trials. Second, the cumulative loss L^* of the best expert does not depend on our predictions p . These two properties allow us to represent the game (2.6) more concisely by a game that is only parametrised by a sequence of actual outcomes x . The game \mathcal{G}_x is defined as follows. To x of length $t = T$ we assign the elementary game with cost

$$\mathcal{G}_x := -L^*(x), \quad (2.7a)$$

while for x of length $t < T$ we set¹

$$\mathcal{G}_x := \begin{array}{c} \begin{array}{cc} & \begin{array}{c} y \\ n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|c|c|} \hline \mathcal{G}_{xy} & \mathcal{G}_{xn} + 1 \\ \hline \mathcal{G}_{xy} + 1 & \mathcal{G}_{xn} \\ \hline \end{array} \end{array} \end{array}. \quad (2.7b)$$

The *regret game with time horizon T* starts without observed outcomes

$$\mathcal{G} := \mathcal{G}_\epsilon. \quad (2.7c)$$

In words, (2.7a) accounts for the cumulative loss of the best expert at the end of the game, while (2.7b) accounts for our mistakes (the +1) directly in the trial where they occur.

Both (2.6) and (2.7) define the regret game with time horizon T . A trivial induction on histories shows that they agree in the following sense.

¹ Incrementing a game \mathcal{G} by $v \in \mathbb{R}$, denoted $\mathcal{G} + v$, means increasing all the costs of its elementary subgames by v . This does not affect the optimal strategy for either player, it simply increases the value of \mathcal{G} by v .

2.5.1. PROPOSITION. For each history $\langle p, x \rangle$ of length $t \leq T$

$$\mathcal{V}_{p,x} = \mathcal{V}_x + L(p, x), \quad \hat{\sigma}_{p,x} = \hat{\sigma}_x \quad \text{and} \quad \hat{\tau}_{p,x} = \hat{\tau}_x,$$

where double subscripts refer to the regret game (2.6), while single subscripts refer to the simplified regret game (2.7).

In particular, the entire games $\mathcal{G}_{\epsilon,\epsilon}$ and \mathcal{G}_ϵ have the same value $\mathcal{V}_{\epsilon,\epsilon} = \mathcal{V}_\epsilon$, since $L(\epsilon, \epsilon) = 0$.

We are now ready to solve the simplified regret game with horizon T . We first solve its subgames locally in Section 2.5.3, then compute our minimax regret in Section 2.5.4, and finally implement our minimax strategy in Section 2.5.5. The next result is rather counter-intuitive.

2.5.3 Expertly Unpredictable Adversarial Outcomes

It follows from the analysis of the one-shot regret game in Section 2.3 that Adversary maximises our expected *cumulative loss* by drawing T outcomes uniformly at random. In the current game however, Adversary strives to maximise our expected *regret*. To this end, Adversary has to be simultaneously unpredictable to make our cumulative loss large and predictable to make the cumulative loss of the best expert small. These goals are obviously in direct conflict. Surprisingly, this discord is resolved optimally by dropping the latter goal. That is, Adversary also maximises our regret by drawing outcomes uniformly at random. This result has been noted earlier, for example in [25, Section 8.3], and holds quite generally. It also drives the minimax analyses [4] and [6].

2.5.2. THEOREM. The game \mathcal{G}_x of (2.7b) has minimax regret \mathcal{V}_x and a unique equaliser saddle point $\langle \hat{\sigma}_x, \hat{\tau}_x \rangle$ where

$$\hat{\sigma}_x = \frac{\mathcal{V}_{xy} - \mathcal{V}_{xn} + 1}{2}, \quad \hat{\tau}_x = \frac{1}{2} \quad \text{and} \quad \mathcal{V}_x = \frac{\mathcal{V}_{xy} + \mathcal{V}_{xn} + 1}{2}.$$

Since $\hat{\tau}_x = 1/2$ independent of the past outcomes x , Adversary's max-min strategy draws outcomes uniformly at random each trial.

Proof. By backwards induction (Section 2.4.1), the solution of \mathcal{G}_x equals the solution of the one-shot game of the values of its constituent games:

$$\mathcal{G}'_x = \begin{array}{c} \begin{array}{cc} & \begin{array}{c} y \\ n \end{array} \\ \begin{array}{c} y \\ n \end{array} & \begin{array}{|c|c|} \hline \mathcal{V}_{xy} & \mathcal{V}_{xn} + 1 \\ \hline \mathcal{V}_{xy} + 1 & \mathcal{V}_{xn} \\ \hline \end{array} \end{array} \end{array}.$$

An easy induction on the length of x shows that $|\mathcal{V}_{xy} - \mathcal{V}_{xn}| \leq 1$. If $|\mathcal{V}_{xy} - \mathcal{V}_{xn}| < 1$, then \mathcal{V}_x has no saddle point in pure strategies since

$$(\mathcal{V}_{xy} - (\mathcal{V}_{xn} + 1))((\mathcal{V}_{xy} + 1) - \mathcal{V}_{xn}) = (\mathcal{V}_{xy} - \mathcal{V}_{xn})^2 - 1 < 0$$

and

$$(\mathcal{V}_{xy} - (\mathcal{V}_{xy} + 1))((\mathcal{V}_{xn} + 1) - \mathcal{V}_{xn}) = -1 < 0$$

verify condition (2.1), so that Lemma 2.2.1 yields the unique equaliser saddle point above. On the other hand if $|\mathcal{V}_{xy} - \mathcal{V}_{xn}| = 1$ then either

$$\mathcal{G}'_x = \begin{array}{c} y \\ n \end{array} \begin{array}{|c|c|} \hline y & n \\ \hline 0 & 0 \\ \hline +1 & -1 \\ \hline \end{array} + \mathcal{V}_{xy} \quad \text{or} \quad \mathcal{G}'_x = \begin{array}{c} y \\ n \end{array} \begin{array}{|c|c|} \hline y & n \\ \hline -1 & +1 \\ \hline 0 & 0 \\ \hline \end{array} + \mathcal{V}_{xn}.$$

In both cases the minimax strategy $\hat{\sigma}_x$ above is a unique, pure equaliser. In the left case, the entire interval $[1/2, 1]$ is maximin, while in the right case the entire interval $[0, 1/2]$ is maximin. In both cases, the unique equaliser maximin saddle point is $\hat{\tau}_x = 1/2$. \square

Theorem 2.5.2 solves the game \mathcal{G}_x in terms of the solutions to its constituent games. We now obtain a direct expression for our minimax regret of the full game \mathcal{V}_ϵ .

2.5.4 Our Minimax Regret

Theorem 2.5.2 shows that, in the worst case, Adversary generates the T actual outcomes x uniformly at random. Then our expected cumulative loss is $T/2$, whatever our strategy. The expected loss of each individual expert is also $T/2$, and neither individual expert has any predictive value. Perhaps paradoxically, the expected cumulative loss of the *best* expert, i.e.

$$\mathbb{E}[L^*(x)] = \sum_{i=0}^T 2^{-T} \binom{T}{i} \min\{T-i, i\},$$

is strictly lower than $T/2$, as can be seen by applying Jensen's inequality to the concave min function. Our minimax regret is exactly the expected amount by which, purely by chance, the best expert beats just guessing.

2.5.3. THEOREM. *Our minimax regret \mathcal{V} in the regret game with time horizon T is sandwiched as follows*

$$\sqrt{\frac{T-1}{2\pi}} \leq \mathcal{V} \leq \sqrt{\frac{T+1}{2\pi}} \quad (2.8)$$

A somewhat cruder asymptotic analysis can be found surrounding [25, Lemma 8.2].

Proof. Recall that our expected cumulative loss is $T/2$. We expand

$$\begin{aligned} \mathcal{V} &= T/2 - \mathbb{E}[L^*(x)] = \sum_{i=0}^T 2^{-T} \binom{T}{i} (i - \min\{T-i, i\}) = \\ & \sum_{i=\lceil T/2 \rceil}^T 2^{-T} \binom{T}{i} (i - (T-i)) = \sum_{i=\lceil T/2 \rceil}^T 2^{-T} T \left(\binom{T-1}{i-1} - \binom{T-1}{i} \right) \\ & \stackrel{\text{telescope}}{=} 2^{-T} T \binom{T-1}{\lceil T/2 \rceil - 1} = 2^{-T} \begin{cases} \frac{T!}{\frac{T-1}{2}! \frac{T-1}{2}!} & \text{if } T \text{ odd,} \\ \frac{T!}{\frac{T}{2}! \frac{T-2}{2}!} & \text{if } T \text{ even.} \end{cases} \end{aligned} \quad (2.9)$$

This elegant telescoping argument goes back to [58]. We now show that

$$\sqrt{\frac{T-1}{2\pi}} < \frac{\Gamma(\frac{T+1}{2})}{\Gamma(\frac{T}{2})\Gamma(\frac{1}{2})} \leq \mathcal{V} \leq \frac{\Gamma(\frac{T}{2}+1)}{\Gamma(\frac{T+1}{2})\Gamma(\frac{1}{2})} < \sqrt{\frac{T+1}{2\pi}}.$$

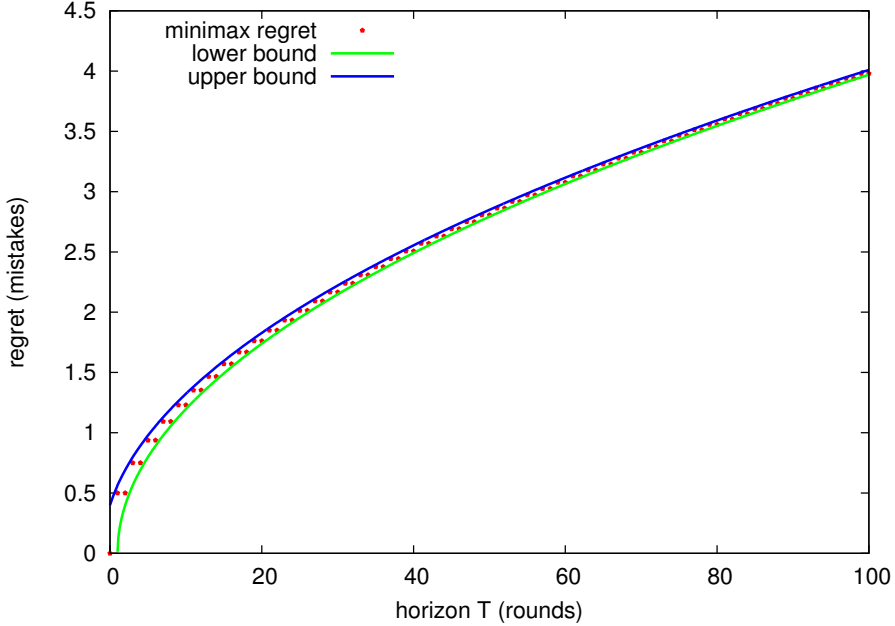
The innermost two inequalities follow from log-convexity of Γ which implies that the expression for odd T in (2.9) is a valid upper bound for even T , whereas the expression for even T in (2.9) is a valid lower bound for odd T . The outermost inequalities use $\Gamma(1/2) = \sqrt{\pi}$, and

$$\frac{\Gamma(x+1/2)}{\Gamma(x)} \frac{\Gamma(x+1)}{\Gamma(x+1/2)} = x,$$

where the smaller left factor must be $< \sqrt{x}$, and the right $> \sqrt{x}$. This implies that $\sqrt{x} < \Gamma(x+1)/\Gamma(x+1/2) < \sqrt{x+1/2}$ for all $x \geq 0$. \square

Figure 2.1 displays our minimax regret in the regret game with time horizon T as a function of T , together with the two bounds of (2.8). We see that the bounds are good approximations even for small T .

Figure 2.1 Minimax regret of the regret game with time horizon T , shown as a function of time horizon T



Our minimax regret \mathcal{V} grows sublinearly in the number of trials T , which means that our minimax strategy $\hat{\sigma}$ indeed learns to predict approximately like the best expert. In particular, when the best expert has small cumulative loss, so do we. Our minimax strategy thus attains the goal set out in the introduction. We now describe how to implement our minimax strategy $\hat{\sigma}$ efficiently.

2.5.5 Executing Our Minimax Strategy

So far, we saw that a strategy achieving the minimax regret \mathcal{V} exists, and we found the recursive expression $\hat{\sigma}_x = (\mathcal{V}_{xy} - \mathcal{V}_{xn} + 1)/2$ in Theorem 2.5.2. In practise, to follow the minimax strategy $\hat{\sigma}_x$, we need an *efficient* method to sample according to it. Unfortunately, the recursive expression does not simplify to a direct expression for $\hat{\sigma}_x$. There are essentially three ways to still implement $\hat{\sigma}_x$.

2.5.5.1 Computing $\hat{\sigma}_x$ Exactly Using the Recursive Expression

Our minimax strategy $\hat{\sigma}_x$ is a function of a sequence of outcomes x , but it is clear that $\hat{\sigma}_x$ only depends on x via the current cumulative loss of both experts. Thus to compute $\hat{\sigma}_x$ for all x , it suffices to compute $\hat{\sigma}_x$ for the $O(T^2)$ possible pairs of cumulative losses of the two experts. This can be done in $O(T^2)$ time using $O(T^2)$ memory. We used this method to compute Figure 2.2, which displays our minimax strategy for time horizon $T = 101$. The colour values indicate the probability of predicting y . The optimal probability of predicting y has a sharp transition from 0 to 1 around the point where both experts have equal cumulative loss.

The advantage of this method is that it computes $\hat{\sigma}_x$ exactly for each x , as required e.g. for graphing. The disadvantage is that the exponent in the running time grows with the number of experts and outcomes, rendering it impractical for problems with many experts. We may give up computing $\hat{\sigma}_x$ exactly, and instead approximate it.

2.5.5.2 Computing $\hat{\sigma}_x$ Approximately

In Theorem 2.5.3 we sandwiched the minimax regret \mathcal{V} tightly around $\sqrt{T/(2\pi)}$. A similar approach can be undertaken to bound the value \mathcal{V}_x of each subgame \mathcal{G}_x , and then approximate $\hat{\sigma}_x$ in terms of these bounds.

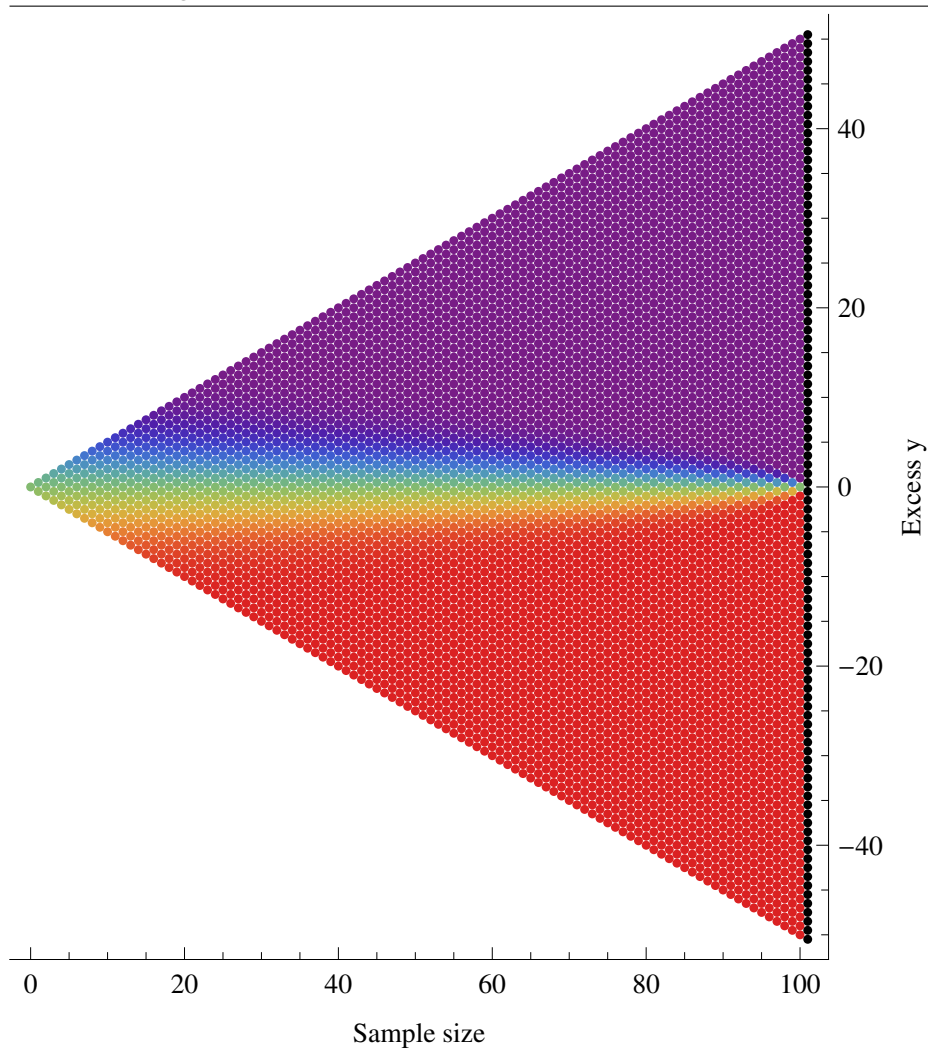
We have not explored this approach in detail, but it is clear that such methods can be much faster than $O(T)$ per trial, and scale better with the number of experts and outcomes. The difficulty lies of course in bounding the additional regret incurred by approximating. There is however a different method, that implements the minimax strategy exactly without computing $\hat{\sigma}_x$ exactly.

2.5.5.3 Not Computing $\hat{\sigma}_x$ At All

Interestingly, following the minimax strategy does not *require* computation of $\hat{\sigma}_x$, it requires predicting y with probability $\hat{\sigma}_x$. We now show that this can be done without computing $\hat{\sigma}_x$. In essence, we can think of our method as performing a randomised exact computation of $\hat{\sigma}_x$.

As proved in Theorem 2.5.2, in game \mathcal{G}_x our minimax strategy predicts y with probability $\hat{\sigma}_x = (\mathcal{V}_{xy} - \mathcal{V}_{xn} + 1)/2$. We now devise a

Figure 2.2 Minimax strategy in the regret game with time horizon $T = 101$. The colour value at position (t, e) indicates the probability $\hat{\sigma}$ that our minimax strategy assigns to observing y next after t trials with $t/2 + e$ occurrences of outcome y and hence $t/2 - e$ occurrences of outcome n . The probabilities range from 0 (red) via $1/2$ (green) to 1 (violet). The game has ended in black dots.



Algorithm 2.1 Minimax algorithm for time horizon T

Input: Game \mathcal{G}_x with observed past outcomes x of length $t < T$.

Sample a continuation z uniformly at random from $\{y, n\}^{T-t-1}$.

Compute $\mathcal{V}_{xyz} = -L^*(xyz)$ and $\mathcal{V}_{xnz} = -L^*(xnz)$.

Predict y with probability $(\mathcal{V}_{xyz} - \mathcal{V}_{xnz} + 1)/2$.

method to generate such predictions without computing \mathcal{V}_{xy} and \mathcal{V}_{xn} exactly. Our algorithm is displayed as Algorithm 2.1. In words, the algorithm takes xy and xn , the two possible one-step extensions of the observed actual outcomes x , completes them both to length T by appending the same random future outcomes z , and plays the outcome whose extension incurs the smaller regret. If both completions incur the same regret, it predicts by flipping a fair coin.

This algorithm is new to the best of our knowledge. It interpolates between the algorithms in [4] and [6], in the sense that it uses random sampling akin to [6], but instead of using a random future to estimate the best expert, it is used to gauge the quality of both outcomes as in the recurrence relations in [4].

We now analyse the algorithm. First we prove correctness, and then consider efficiency.

2.5.4. THEOREM. *Algorithm 2.1 implements our minimax strategy $\hat{\sigma}$.*

Proof. By Theorem 2.5.2, for any history x of length t and z sampled uniformly at random from $\{y, n\}^{T-t}$, we have $\mathbb{E}[\mathcal{V}_{xz}] = (T-t)/2 + \mathcal{V}_x$. Since the sequences xyz and xnz differ only in one position, the cumulative loss of the best expert differs by at most one, so that $|\mathbb{E}[\mathcal{V}_{xyz}] - \mathbb{E}[\mathcal{V}_{xnz}]| \leq 1$. The probability that the algorithm predicts y is therefore given by

$$\mathbb{E}\left[\frac{\mathcal{V}_{xyz} - \mathcal{V}_{xnz} + 1}{2}\right] = \frac{\mathbb{E}[\mathcal{V}_{xyz} - \mathcal{V}_{xnz}] + 1}{2} = \frac{\mathcal{V}_{xy} - \mathcal{V}_{xn} + 1}{2} = \hat{\sigma}_x$$

as desired. □

Performing this procedure in trial t takes $O(T-t)$ steps and uses $O(T-t)$ memory. Predicting T outcomes thus takes $O(T^2)$ time in total. Since the memory can be reused, $O(T)$ memory suffices.

The advantage of this method is that it is extremely simple, and that it scales to more difficult prediction problems extremely well, as will be explored in Section 2.8. For our two constant experts, the effect of appending a random future is that the current cumulative losses of both experts are incremented by a binomial random variable. Drawing a binomial random variable can be done more efficiently, see e.g. [87].

Another interesting possibility is to re-use a single randomly drawn full future of length T , reducing the running time to $O(T)$ in total. This makes our predictions in subsequent trials statistically dependent, and this can be used against us by Adversary. However, for so-called *oblivious adversaries* [25], i.e. adversaries that do not look at our predictions at all, this fast strategy still attains the minimax regret.

2.6 Variations on a Theme

We modelled predicting a binary outcome as the regret game with time horizon T , quantified the best achievable regret, and implemented our minimax strategy efficiently. We now extend our results to a variety of similar prediction problems, simultaneously obtaining near-optimal solutions to the corresponding regret games, and resulting in new insights into our minimax strategy.

2.6.1 Stopping Early

We now consider what happens if we give Adversary the power to stop the game early, before the time horizon T is reached. That is, at the beginning of each trial, we allow Adversary to declare the game finished, and if she does we suffer the regret at that time. That is, if Adversary stops in history $\langle p, x \rangle$, we suffer $L(p, x) - L^*(x)$. If Adversary chooses to continue, then the game proceeds as before. We now show that this extra power cannot be used to beat Theorem 2.5.3.

2.6.1. THEOREM. *Our minimax regret without and with adversarial stopping are identical.*

Proof. We show that stopping is a dominated strategy. Say that we are in some history $\langle p, x \rangle$, and assume w.l.o.g. that constant- y is the best expert. By stopping, Adversary inflicts regret $L(p, x) - L^*(x)$. However, she is better off by playing outcome n until time T . This will not

change the cumulative loss of the best expert, nor will it decrease our cumulative loss. Hence our eventual regret is at least our current regret. Thus stopping is never useful. \square

Note that we do not claim that our minimax regret (2.8) now can be evaluated with the actual stopping time substituted for the original time horizon T . We simply say that our minimax regret with the original time horizon T remains unchanged. We now obtain a bound that can be evaluated on the actual stopping time.

2.6.2 Unknown Time Horizon T

It is not always realistic to assume that the number of trials T is known beforehand. Still, it is possible to approximately achieve the minimax regret (2.8) without knowing T by a method called the *doubling trick* [25]. Start by segmenting the trials as follows.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	-----

Play the first trial as if $T = 1$. Then reset the algorithm and play the next two trials as if $T = 2$. Again, reset the algorithm and play the subsequent four trials as if $T = 4$ etc. Each reset completely restarts the algorithm, ignoring the history generated until then. This procedure guarantees that in segment i (of length 2^{i-1}), our regret is at most $\sqrt{(2^{i-1} + 1)/(2\pi)}$ w.r.t. the best expert on that segment. The cumulative loss of the segmentwise best expert is at most the cumulative loss of the best expert. To analyse the regret, we first prove the following intermediate result.

2.6.2. LEMMA. *For any natural number b*

$$\sum_{i=1}^b \sqrt{2^{i-1} + 1} \leq (1 + \sqrt{2})\sqrt{2^b}.$$

Proof. By concavity of the square root $\sqrt{x+1} \leq \sqrt{x} + \frac{1}{2\sqrt{x}}$, so that

$$\begin{aligned} \sum_{i=1}^b \sqrt{2^{i-1} + 1} &\leq \sum_{i=1}^b \left(\sqrt{2^{i-1}} + \frac{1}{2\sqrt{2^{i-1}}} \right) = \frac{\sqrt{2^b} - 1}{\sqrt{2} - 1} + \frac{1 - \sqrt{2^{-b}}}{2 - \sqrt{2}} \\ &= (1 + \sqrt{2}) \left(\sqrt{2^b} - 1 + 1/\sqrt{2} - \sqrt{2^{-b}}/\sqrt{2} \right). \end{aligned}$$

Then observe that $-1 + 1/\sqrt{2} - \sqrt{2^{-b}}/\sqrt{2} \leq 0$ for all b . \square

This lemma allows us to analyse the guarantee provided by the doubling trick. If the actual length is T , we use $\lceil \log_2(T+1) \rceil$ segments. The last segment may be stopped early, but by the previous section the full regret bound still holds there. In total, we thus guarantee regret

$$\sum_{i=1}^{\lceil \log_2(T+1) \rceil} \sqrt{\frac{2^{i-1} + 1}{2\pi}} \leq (1 + \sqrt{2}) \sqrt{2^{\lceil \log_2(T+1) \rceil}} \leq (2 + \sqrt{2}) \sqrt{\frac{T+1}{2\pi}}$$

uniformly over time. The first inequality is Lemma 2.6.2, and the second uses $\lceil x \rceil \leq x + 1$.

This bound shows that not knowing the time horizon T multiplies the regret (2.8) by at most a factor $2 + \sqrt{2} \approx 3.414$. This is close to the slightly better factor $2\sqrt{\pi \ln(2)} \approx 2.951$ that is achieved by the exponentially weighted average forecaster with suitably decreasing learning rate [25, Theorem 2.3].

2.6.3 Two Adversarial Experts

In this section we replace our two constant experts with two adversarial or black-box experts (see the taxonomy in Section 1.6), whose predictions are determined by Adversary. It is clear that this increases Adversary's power, and hence increases our minimax regret. We now show that our regret remains as it is, that is, constant experts are the worst case.

First observe that trials in which the experts agree are wasted by Adversary. Namely, in such trials we can ensure that our regret does not grow, by issuing the same prediction as both experts. Since such trials do not influence our regret, Adversary may as well delay them to the end of the game. But then they amount to early stopping, which we proved suboptimal in Section 2.6.1.

Second, observe that we can interchange the labels y and n (by flipping both predictions and outcomes) while preserving the instantaneous loss of each prediction/outcome pair. Hence we may assume that one expert always predicts y , while the other always predicts n .

Combining disagreement and label interchange, we see that our original two constant experts are actually adversarial, and that the minimax algorithm for constant experts can be transformed into the minimax algorithm for adversarial experts by reinterpreting $\hat{\sigma}_x$ as the probability of following the first expert.

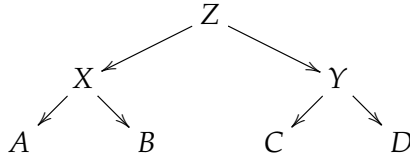
2.6.4 Many Experts

We have an algorithm for combining the binary predictions of two experts. Faced with many experts, an obvious way to combine their (still binary) predictions is to combine them in a binary tree.

2.6.3. THEOREM. *The algorithm that combines n experts recursively in a binary tree, playing the minimax strategy for adversarial experts at each internal node, guarantees expected regret at most*

$$\lceil \log(n) \rceil \sqrt{\frac{T+1}{2\pi}}.$$

Proof. We prove the simple case of four experts. Let's call them A, B, C and D . We recursively combine them as follows



Here the leaves A, B, C and D are basic experts, and the internal meta-experts X, Y and Z represent experts that predict by running the minimax algorithm on their respective children. To analyse the regret of Z w.r.t. the best basic expert, we abbreviate the cumulative loss of each expert $\xi \in \{A, B, C, D, X, Y, Z\}$ by L_ξ , and denote the regret incurred by the meta-experts X, Y and Z w.r.t. their children by R_X, R_Y and R_Z . The cumulative losses and regrets of the meta-experts are random variables, determined by the internal randomisation of the minimax algorithm. We have

$$\begin{aligned} \mathbb{E}[L_Z] &= \mathbb{E}[\min\{L_X, L_Y\} + R_Z] \\ &= \mathbb{E}\left[\min\{\min\{L_A, L_B\} + R_X, \min\{L_C, L_D\} + R_Y\}\right] + \mathbb{E}[R_Z] \\ &\leq \min\{\min\{L_A, L_B\} + \mathbb{E}[R_X], \min\{L_C, L_D\} + \mathbb{E}[R_Y]\} + \mathbb{E}[R_Z] \\ &\leq \min\{L_A, L_B, L_C, L_D\} + 2\sqrt{\frac{T+1}{2\pi}} \end{aligned}$$

by applying Jensen's inequality to the concave function \min , and using the minimax regret bound Theorem 2.5.3. The case for n experts is analogous. \square

Expected regret that depends logarithmically on the number of experts n is already quite good, but even better guarantees can be obtained. The exponentially weighted average forecaster [25, Theorem 2.2] achieves expected regret bounded by

$$\sqrt{\frac{T}{2} \ln(n)},$$

with square-root-of- \ln dependence on the number of experts n .

2.7 Good Best Expert

Let's consider again the simplest case, prediction with two constant experts. Our minimax strategy for the regret game with time horizon T guarantees the minimax regret $\sqrt{(T+1)/(2\pi)}$ w.r.t. the best expert, irrespective of Adversary's strategy. In fact, our minimax strategy equalises the expected regret over all possible strategies for Adversary. That is, we suffer the same expected regret whatever the cumulative loss of the best expert. There are no sequences of outcomes for which we are *lucky* in the sense that our expected regret is actually smaller than the bound indicates.

The concept of *luckiness* [163, 71] is the idea that we desire to incur tiny regret whenever the best expert has small cumulative loss, and to achieve this, are willing to accept a slight increase in overhead when the best expert has moderate or large cumulative loss.

In this section we investigate a fundamentally different strategy, namely the strategy that enforces maximal luck. To this end, we first assume that we know that the best expert makes at most k mistakes. We then create a new prediction game where this assumption is built into the rules, i.e. constrains Adversary's moves. As before, we find the minimax strategy in this prediction game. Finally, we get rid of this assumption using the doubling trick (as in Section 2.6.2).

The constraint that the cumulative loss of the best expert is at most k serves as a "loss horizon", replacing the role of the time horizon T . The minimax strategy and regret that we obtain in this section are therefore parametrised by k instead of T .

2.7.1 The Regret Game With Loss Horizon k

The *regret game with loss horizon k* is defined as follows. In trial t , we pick a prediction p_t and Adversary plays an actual outcome x_t . As before, both moves are randomised independently. If in any history $\langle p, x \rangle$ the best expert has cumulative loss $L^*(x) > k$, the game ceases and we suffer regret $-\infty$. This is so bad for Adversary that she will never let this happen. Otherwise, the game continues for infinitely many trials. In the resulting infinite history $\langle p, x \rangle$ we suffer the regret $L(p, x) - L^*(x)$. We may allow Adversary to stop the game early, but this is a dominated strategy that does not increase her power, by the same reasoning as in Section 2.6.1. We first simplify the game, then solve it.

2.7.2 Simplified Regret Game

Infinite plays of the regret game with loss horizon k are pretty boring apart from a finite initial part of length at most $2k + 1$. This is because after at most $2k + 1$ outcomes, one expert must have cumulative loss $k + 1$, meaning that this expert cannot be the eventual best expert. We say that such an expert is *dead*.

We now show that our minimax strategy never follows a dead expert. This in turn means that once an expert dies, the regret never changes. We and the best expert either both or neither incur loss.

Say w.l.o.g. that the constant- n expert is dead. Then both our cumulative loss and the cumulative loss of the best expert decompose over trials. That is, each such trial we are facing the game that changes our regret as follows:

	y	n
y	0	0
n	+1	-1

This game has a saddle point in pure strategies, $\hat{\sigma} = \hat{\tau} = 1$, and game value $\mathcal{V} = 0$. Again $\hat{\sigma}$ is an equaliser strategy: by deterministically playing y , we guarantee that our eventual regret equals our current regret $L(p, x) - L^*(x)$, irrespective of Adversary's strategy.

The regret game with loss horizon k is quite different from the regret game with time horizon T . Surprisingly, we now solve the former game problem by reducing it to the latter.

2.7.3 Reduction to Time Horizon $T = 2k + 1$

We argued that once an expert is dead, our minimax strategy ensures that the eventual regret equals the current regret. So we might as well stop the game then or at any later time. We propose to stop the regret game with loss horizon k at time $T = 2k + 1$. We know that at this time exactly one expert is dead, so the game can safely be stopped. Note in particular that the rule that hands out $-\infty$ regret when both experts die cannot yet have been invoked. Vice versa, in the regret game with time horizon $T = 2k + 1$, the best expert will suffer loss at most k . But this means that these games are identical, and hence share the same value and optimal strategies. In particular

2.7.1. THEOREM. *The regret game with loss horizon k has minimax regret*

$$\sqrt{\frac{k}{\pi}} \leq \mathcal{V} \leq \sqrt{\frac{k+1}{\pi}}. \quad (2.10)$$

Adversary's maximin strategy samples outcomes uniformly at random until one expert dies. After that, she always chooses the outcome predicted by the live expert. When both experts are still alive, our minimax strategy is the one described in Section 2.5.5. After one expert has died, we follow the live expert.

Proof. By Theorem 2.5.3. □

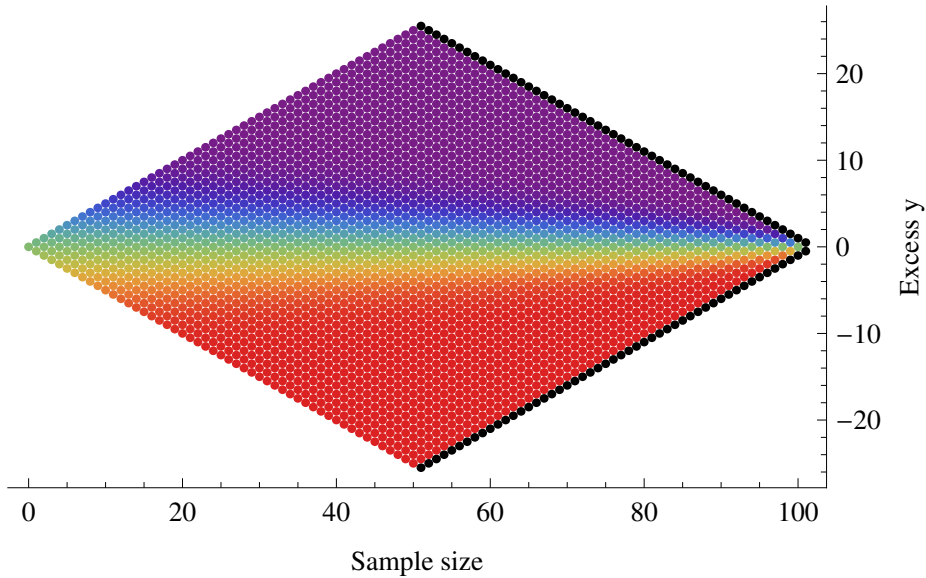
The equivalence of a loss horizon with a particular time horizon has not been noted in the literature, perhaps because it is most apparent for two experts. A generalisation to many experts can undoubtedly be obtained, but falls outside the scope of this minimalistic introduction.

Our minimax strategy is displayed for $k = 50$ in Figure 2.3. This figure is a truncated version of the optimal strategy with time horizon $T = 2k + 1 = 101$, which is shown in Figure 2.2.

2.7.4 Unknown Loss Horizon k

In Section 2.6.2 we eliminated the need to know the time horizon T by using the doubling trick. Here we face the problem that we may not know k . Again we apply the doubling trick. We run our minimax algorithm assuming that $k = 1$. When the worst expert dies, the game with this assumption is essentially over, and we restart completely, now

Figure 2.3 Minimax strategy in the regret game with loss horizon $k = 50$. This is a subplot of Figure 2.2. The colour value at position (t, e) indicates the probability $\hat{\sigma}$ that our minimax strategy assigns to observing y next after t trials with $t/2 + e$ occurrences of outcome y and hence $t/2 - e$ occurrences of outcome n . The game has (essentially) ended in black dots, since one expert is dead.



with $k = 2$. Again when the worst expert dies we restart with doubled loss horizon. The regret guarantee becomes

$$\sum_{i=1}^{\lceil \log_2(k+1) \rceil} \sqrt{\frac{2^{i-1} + 1}{\pi}} \leq (1 + \sqrt{2}) \sqrt{2^{\lceil \log_2(k+1) \rceil}} \leq (2 + \sqrt{2}) \sqrt{\frac{k+1}{\pi}}.$$

The first inequality is Lemma 2.6.2, and the second uses $\lceil x \rceil \leq x + 1$.

This bound shows that not knowing the loss horizon k multiplies the regret (2.10) by at most a factor $2 + \sqrt{2} \approx 3.414$.

We now leave our two constant experts, to compete with more complicated white-box experts.

2.8 Competing with a 1-Lipschitz Best Expert

We now consider prediction with more complicated white-box experts (see Section 1.6). In fact, we abstract so dramatically that the experts almost disappear. Fix a time horizon T . We take as our basic ingredient a function $L^*: \{y, n\}^T \rightarrow \mathbb{R}$ that measures the cumulative loss of the best expert.

The L^* -regret game with time horizon T is obtained by substituting the given function L^* into defining equation (2.7) on page 30. That is, the game structure remains the full binary tree of depth T , but the payoff is now the regret w.r.t. the best expert as specified by L^* .

In this section, we assume that L^* is 1-Lipschitz in the Hamming metric, which means that $L^*(x)$ changes by at most 1 when we flip one outcome of x . This assumption obviously holds for all L^* that we have considered so far, since the cumulative loss of *any* static (see Section 1.6) expert changes by at most one if a single outcome is flipped, and hence so does the cumulative loss of the best expert.

By choosing an intricate best expert cumulative loss function L^* , we obtain interesting prediction games. The many-expert bounds and algorithms that we saw in Section 2.6.4 are tight for adversarial experts, that is, assuming that Adversary can control the loss of each expert independently, or at least approximately so. But the experts underlying L^* can be quite far from independent, implying that the minimax algorithm is fundamentally different, and guarantees smaller regret than the adversarial bounds suggest.

We will consider the particular application of switching experts (or tracking) in Section 2.9. Interestingly and curiously, it is possible to obtain a single minimax strategy, parametrised by L^* , that works for all 1-Lipschitz L^* .

2.8.1 The Minimax Strategy

We now show that Algorithm 2.1 remains our minimax strategy in the L^* -regret game with time horizon T . As far as we know this unexpected result is new. We also show that, as before, Adversary's maximin strategy draws outcomes uniformly at random. The crucial observation is that the solution of the vanilla regret game with time horizon T , given as Theorem 2.5.2, applies to the more general L^* -regret game.

2.8.1. THEOREM. Fix a 1-Lipschitz function L^* , time horizon T , and let \mathcal{G} be the L^* -regret game with time horizon T as defined in equation (2.7) on page 30. Then for any sequence of outcomes x of length $t < T$, the game \mathcal{G}_x has minimax regret \mathcal{V}_x and a unique equaliser saddle point $\langle \hat{\sigma}_x, \hat{\tau}_x \rangle$ where

$$\hat{\sigma}_x = \frac{\mathcal{V}_{xy} - \mathcal{V}_{xn} + 1}{2} \quad \hat{\tau}_x = \frac{1}{2} \quad \mathcal{V}_x = \frac{\mathcal{V}_{xy} + \mathcal{V}_{xn} + 1}{2}$$

Again, Adversary's maximin strategy is to play uniformly at random. Note that these recurrence relations are exactly the same as obtained in Theorem 2.5.2. However, the different base case $\mathcal{V}_x = -L^*$ for final x of length T , affects all derived values and minimax strategies.

Proof. The critical inequality is to show $|\mathcal{V}_{xy} - \mathcal{V}_{xn}| \leq 1$, the remainder of the proof equals that of Theorem 2.5.2. We prove this inequality by induction on the length t of x . With z denoting a sequence of $T - t - 1$ outcomes drawn uniformly at random,

$$\mathcal{V}_{xy} - \mathcal{V}_{xn} = \mathbb{E}[\mathcal{V}_{xyz} - \mathcal{V}_{xnz}] = \mathbb{E}[-L^*(xyz) + L^*(xnz)].$$

By Jensen's inequality applied to the convex absolute-value function, and by using that L^* is 1-Lipschitz we obtain

$$|\mathcal{V}_{xy} - \mathcal{V}_{xn}| \leq \mathbb{E}[|-L^*(xyz) + L^*(xnz)|] \leq 1. \quad \square$$

It follows that Algorithm 2.1 implements our minimax strategy. The efficiency of this algorithm now depends on the resources required to evaluate L^* . It also follows that the game value equals

$$\mathcal{V} = T/2 - \mathbb{E}[L^*(x)].$$

We now consider a simple example.

2.8.2 Example

To illustrate Theorem 2.8.1, we now work through a simple example. We consider predicting $T = 3$ outcomes, and compete with all experts that predict y exactly once. That is, our goal is to learn which outcome is the y . In formulas, our set of experts is

$$\mathbb{E} := \{ynn, nyn, nny\},$$

so that the cumulative loss of the best expert is

$$L^*(x) := \min_{\zeta \in \Xi} L(\zeta, x).$$

The function L^* is clearly 1-Lipschitz, since the cumulative loss of *any* fixed sequence of predictions changes by at most one if a single outcome is flipped, and hence so does the cumulative loss of the best expert. We call this particular regret game the Ξ -regret game.

Figure 2.4 displays the solution, obtained by backwards induction (Section 2.4.1), to the Ξ -regret game. The figure indicates the value \mathcal{V}_x of each subgame \mathcal{G}_x , and either the best expert(s) when the game is over or the minimax strategy $\hat{\sigma}_x$ when trials remain. We see that the minimax regret of the full game is $3/4$, and that this equals our expected cumulative loss, which is $T/2 = 3/2$, minus the expected cumulative loss of the best expert, which is $(2 + 1 + 1 + 0 + 1 + 0 + 0 + 1)/8 = 3/4$.

We see that, starting at $\hat{\sigma} = 1/4$, our minimax strategy doubles the probability of predicting y as long as n s are observed. This is reminiscent of the investment strategy underlying the St. Petersburg paradox. After the first y , an eventual best expert is revealed, and we deterministically predict n .

This was a simple example. An important choice of L^* arises from considering as experts all prediction sequences with few switches. We now consider this application.

2.9 Switching

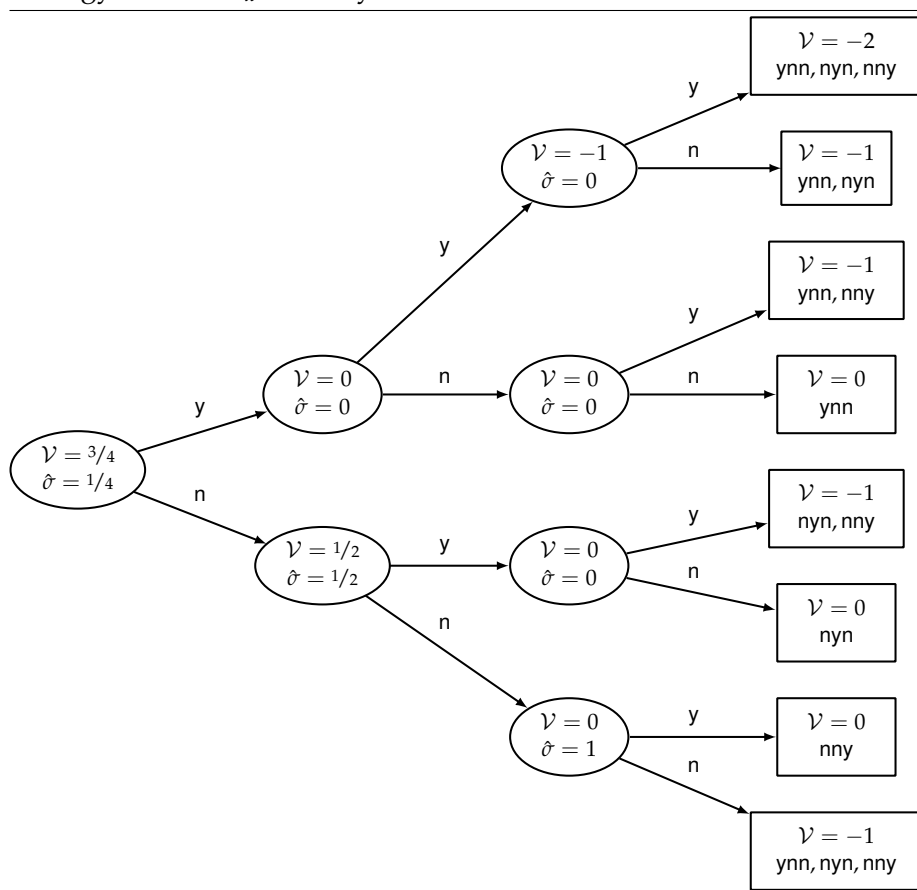
In this section, we apply the theory of competing with a 1-Lipschitz best expert to prediction with experts that divide the outcomes in blocks, predict constantly within blocks, and flip their prediction between consecutive blocks. An example such expert ζ predicts

$$\zeta = \underbrace{yyyyy}_{\text{Block 1}} \underbrace{nnnnnnnn}_{\text{Block 2}} \underbrace{yyyy}_{\text{Block 3}}.$$

Switch 1 Switch 2
 \downarrow \downarrow

We may think of such switching experts either as static experts that switch between two *predictions* (c.f. Section 1.6) or equivalently as meta-experts that switch between the two constant *experts* (c.f. Section 1.9).

Figure 2.4 Backwards induction solution to the $\bar{\epsilon}$ -regret game. For each sequence x of outcomes we have computed the value \mathcal{V}_x of the subgame \mathcal{G}_x . Squares display the best expert(s) for x of length $t = T$. Ellipses give our minimax strategy $\hat{\sigma}_x$ for x of length $t < T$. Adversary's maximin strategy satisfies $\hat{\tau}_x = 1/2$ by Theorem 2.8.1.



The number of *blocks* always exceeds the number of *switches* by one, since switches occur between blocks. Whereas competing with the best constant expert allows us to learn the global trend of the sequence of outcomes, competing with switching experts allows us to track local trends. This is useful in applications where we expect the outcome-generating process to infrequently change, for example because it can be in several states. Our goal is to obtain an efficient minimax strategy.

The sets of prediction sequences of length T with *exactly* and *at most* m blocks are defined by

$$\mathbb{S}_T^m := \left\{ \zeta \in \{y, n\}^T \mid \sum_{1 \leq t < T} \mathbf{1}_{\zeta_t \neq \zeta_{t+1}} = m - 1 \right\} \quad \text{and} \quad \mathbb{S}_T^{\leq m} := \bigcup_{i \in [m]} \mathbb{S}_T^i.$$

Our example expert ζ above is a member of \mathbb{S}_{16}^3 and of $\mathbb{S}_{16}^{\leq m}$ for all $m \geq 3$. The numbers of such experts with exactly and at most m blocks equal

$$|\mathbb{S}_T^m| = 2 \binom{T-1}{m-1} \quad \text{and} \quad |\mathbb{S}_T^{\leq m}| = 2 \sum_{i \in [m]} \binom{T-1}{i-1}.$$

We now compete with the rather large set $\mathbb{S}_T^{\leq m}$. The cumulative loss of the best expert in this set is given by

$$L^*(x) := \min_{\zeta \in \mathbb{S}_T^{\leq m}} L(\zeta, x)$$

The function L^* is obviously 1-Lipschitz, since the cumulative loss of *any* sequence of predictions changes by at most one if a single outcome is flipped, and hence so does the cumulative loss of the best expert in $\mathbb{S}_T^{\leq m}$.

The *switching regret game with time horizon T* is obtained by substituting this particular L^* into equation (2.7). The set $\mathbb{S}_T^{\leq m}$ is rather large, and hence computing the cumulative loss of the best expert by iterating over its members quickly becomes impractical, both in T and m . Fortunately, the set $\mathbb{S}_T^{\leq m}$ is highly regular, and its structure can be used to evaluate L^* , allowing us to efficiently execute our minimax strategy.

2.9.1 Executing Our Minimax Strategy

We saw in Section 2.8 that we can follow our minimax strategy once we have an efficient method to evaluate $L^*(x)$, the cumulative loss of the

best expert on outcomes x . In case of switching, the cumulative loss of the best expert can be computed efficiently by dynamic programming. The trick is to recursively define $L_y^m(x)$, the cumulative loss on outcomes x of the *best* sequence of predictions with at most m blocks that predicts y next. The recursive clauses are, for $m > 1$

$$\begin{aligned} L_y^m(xy) &:= L_y^m(x) & L_y^m(xn) &:= \min\{L_y^m(x) + 1, L_n^{m-1}(x)\} \\ L_n^m(xn) &:= L_n^m(x) & L_n^m(xy) &:= \min\{L_n^m(x) + 1, L_y^{m-1}(x)\} \end{aligned}$$

and the base cases are

$$\begin{aligned} L_y^m(\epsilon) &:= 0 & L_y^1(xy) &:= L_y^1(x) & L_y^1(xn) &:= L_y^1(x) + 1 \\ L_n^m(\epsilon) &:= 0 & L_n^1(xn) &:= L_n^1(x) & L_n^1(xy) &:= L_n^1(x) + 1 \end{aligned}$$

An easy induction on both m and the length of x shows that

2.9.1. PROPOSITION. *Fix $1 \leq m \leq T$. For each sequence x of T outcomes*

$$L^*(x) = \min\{L_y^m(x), L_n^m(x)\}.$$

Evaluating $L^*(x)$ for a history of length T can hence be done in time $O(mT)$. Predicting T outcomes using the minimax Algorithm 2.1 with this method of evaluating L^* takes $O(mT^2)$ time in total, and uses $O(m)$ memory.

2.9.2 Minimax Regret

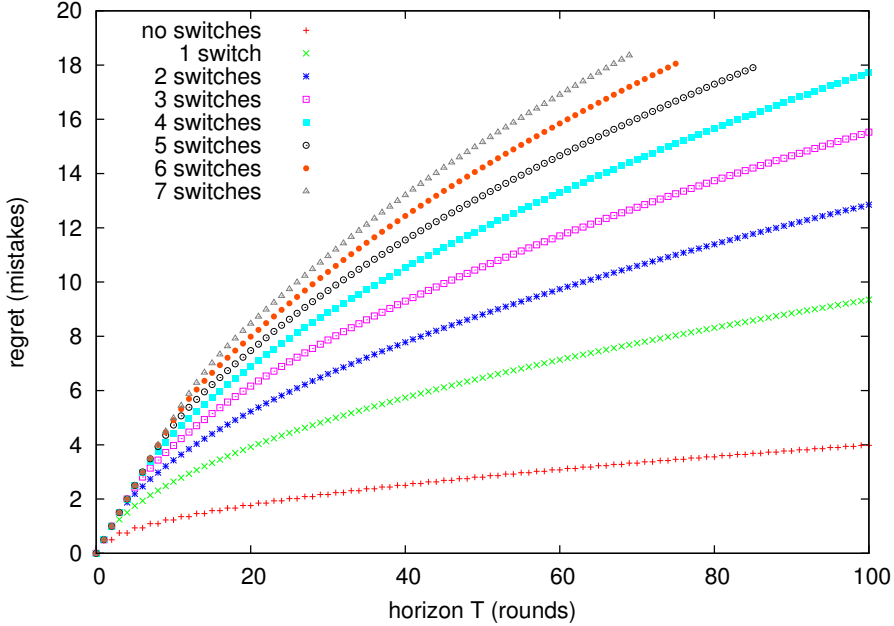
To evaluate our minimax regret, we use the fact that Adversary's maximin strategy, which generates outcomes uniformly at random, is an equaliser strategy. Our expected regret is thus

$$\mathcal{V} = T/2 - \mathbb{E}[L^*(x)]$$

irrespective of the strategy that we follow. The difficulty is, of course, to evaluate the expectation.

Using a trick similar to the dynamic programming solution of Section 2.9.1, we can evaluate our minimax regret \mathcal{V} exactly for particular block counts m and time horizons T , resulting in Figure 2.5. For $m = 1$ blocks there are no switches, and our minimax regret is $\sqrt{T/(2\pi)}$ as before. We see that each additional block increases our minimax regret by less.

Figure 2.5 Minimax regret of the regret game with 0–5 switches as a function of time horizon T



Adversarial Upper Bound The power of Adversary increases when we give her $|\mathcal{S}_T^{\leq m}|$ many adversarial experts. Hence, our minimax regret is bounded by the fully adversarial bound of [25, Theorem 2.3], yielding

$$\sqrt{T/2 \ln |\mathcal{S}_T^{\leq m}|}$$

Since $|\mathcal{S}_T^m| \leq (T-1)^{m-1}$, we have

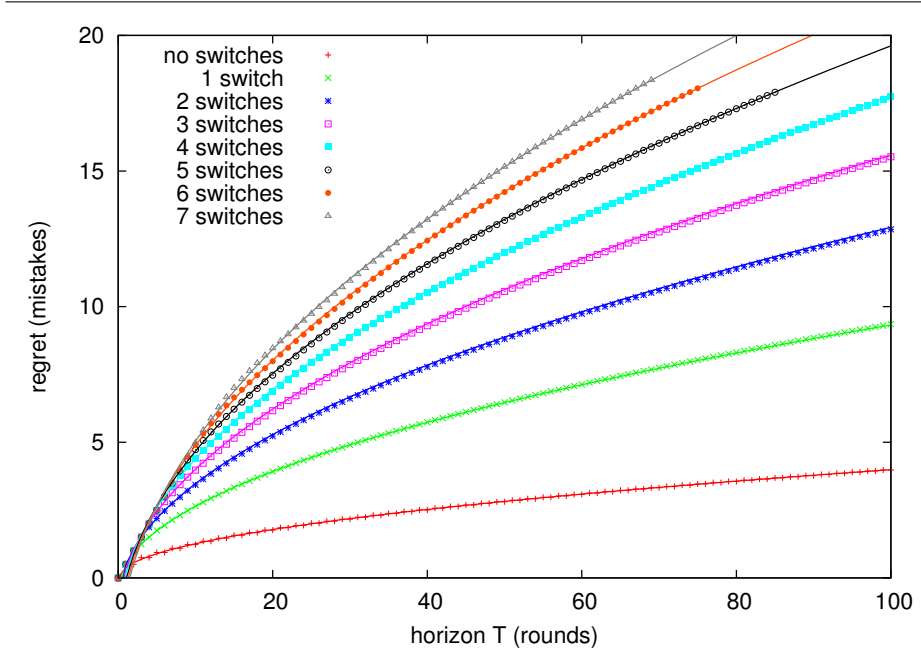
$$|\mathcal{S}_T^{\leq m}| \leq 2 \sum_{i=0}^{m-1} (T-1)^i = 2 \frac{(T-1)^m - 1}{(T-1) - 1} \leq 2T^{m-1}.$$

So that when $T \geq 2$, the minimax regret is bounded by

$$\sqrt{T/2 \ln |\mathcal{S}_T^{\leq m}|} \leq \sqrt{T/2m \ln(T)}. \quad (2.11)$$

By construction, this bound is too pessimistic. We now quantify the discrepancy.

Figure 2.6 Empirical Fit, equation (2.12), overlaid on Figure 2.5. For $m = 1$ (shown in red), we have graphed the minimax regret.



Empirical Fit Looking at Figure 2.5, it seems that the minimax regret is well approximated, for all $m \geq 2$, by

$$\sqrt{6(m-1)\frac{T}{2\pi}} - \frac{m-1}{\sqrt{5}}. \quad (2.12)$$

Figure 2.6 displays the fitted equation (2.12) overlaid on the exact minimax regrets from Figure 2.5. Indeed, we see that the fit is highly accurate. This suggests that (2.11) may be improved, getting rid of the logarithmic dependence on T under the square root.

We are in the curious situation that we have the optimal algorithm, but it is hard to quantify its guarantees. Deriving a better analytic bound is left as an open problem.

2.10 Related Research

We review the literature on online learning with the $0/1$ loss.

Algorithm 2.2 Minimax algorithm for loss horizon k

Input: Game \mathcal{G}_x with outcome sequence x .

Draw samples $z = z_1, z_2, \dots$ uniformly at random from $\{y, n\}$ until the worst expert on xz dies, i.e. reaches cumulative loss $k + 1$.

Then follow the other, live expert.

Prediction Problems The early work on sequential prediction with $0/1$ loss includes the Weighted Majority algorithm [108] for combining the binary advice of n adversarial experts and the Aggregating algorithm [181]. These algorithms have a *learning rate* that needs to be supplied, and which must be tuned based on knowledge of time horizon T or loss horizon k . Uniform bounds can be obtained by the doubling trick, or more sophisticatedly, by decreasing the learning rate incrementally, as analysed e.g. by [25] for the exponentially weighted average forecaster.

A game-theoretic optimal solution for prediction with adversarial experts with loss horizon k was found much later in the form of the Binning algorithm by [4].

Decision Problems In prediction problems, experts suffer loss because they issue incorrect predictions. In the more abstract decision problems the predictions disappear, and experts are entities that can be followed and that suffer loss. Freund and Schapire adapted Weighted Majority to this setting, yielding the Hedge algorithm [59], and proved the regret bound we presented in Section 1.5.1 of the introduction Chapter 1.

The minimax algorithm for the decision problem with loss horizon k was recently obtained by Abernethy, Warmuth and Yellin in [6]. For comparison, we have displayed this algorithm, specialised to predicting two outcomes with our two constant experts, as Algorithm 2.2. This algorithm is also minimax for prediction with two constant experts, because two constant experts are essentially adversarial by the reasoning explained in Section 2.6.3. Both Algorithm 2.1 and Algorithm 2.2 sample a single sequence of random outcomes until the game is over. However, this outcome is used in a fundamentally different way. Algorithm 2.1 uses it to gauge the quality of both possible *predictions*. On the other hand, Algorithm 2.2 uses it estimate the eventual *best expert*.

Bandit Problems In the even more abstract bandit problems [10, 118, 176, 155, 5, 26, 24, 50, 55, 143, 158], the loss feedback disappears, and we only observe the loss of the expert we choose to follow, or of the action that we choose to play. This leads to the problem of *exploration vs exploitation*. To identify the best expert, we need to follow (exploration) all experts once in a while, to monitor their quality. But to suffer small loss, we want to follow (exploitation) the best expert exclusively. The challenge in bandit problems is to interleave these two goals, trading off the benefits of both.

Absolute Loss The *absolute loss* of a randomised prediction is defined as its expected 0/1 loss [25]. Prediction with 0/1 loss and with absolute loss are hence tightly related: their minimax regrets and strategies are identical, modulo one important and subtle difference. To issue minimax predictions under 0/1 loss, it suffices to *randomly* compute $\hat{\sigma}$, the optimal probability of predicting y . For absolute loss it needs to be computed *exactly*. This means that Algorithm 2.1 cannot be used for absolute loss.

2.11 Conclusion

We presented and analysed four prediction games. We first considered predicting a single binary outcome with the help of two constant experts. We showed that the corresponding one-shot regret game has minimax regret $1/2$, which we can achieve by predicting uniformly at random. We then proceeded to predicting a sequence of binary outcomes. We first considered competing with the best constant expert for T trials. In the corresponding regret game with time horizon T our minimax regret essentially equals $\sqrt{T/(2\pi)}$. We then considered competing with the best constant expert given that the best expert makes at most k mistakes. In the regret game with loss horizon k , our minimax regret essentially equals $\sqrt{k/\pi}$. We concluded by competing with the best expert that switches at most $m - 1$ times. In each case, we gave an efficient randomised algorithm for playing the minimax strategy.

The prediction problems that we considered form just the tip of the iceberg of online learning, and many interesting problems readily suggest themselves.

1. Prediction problems with more than two outcomes.
2. Decision problems (without outcomes).
3. More than two experts.
4. More complicated white-box experts.
5. Black-box (adversarial) experts.
6. More complicated (gray-box) meta-experts, e.g. that
 - (a) switch between all of the above
 - (b) combinatorially combine all of the above
7. Competing with gray-box experts with loss horizon k .
8. All the above for loss functions other than the 0/1 loss.

In the remainder of this dissertation we consider the following specific problems. In Chapter 3 we introduce a general graphical framework for the construction of meta-experts for log loss, with a focus on switching. In Chapter 4 we extend this framework to switching between gray-box experts. In particular, this allows us to switch between learning experts. Then in Chapter 5 we introduce a novel way to switch between two experts, motivated by finance but generally applicable to decision problems with log-loss or log-return. Finally, in Chapter 6 we consider combinatorial combinations of experts for 0/1 loss and/or absolute loss.

The approach taken in future chapters is different from the method developed in this chapter. The decision problems considered there are more complicated, and hence finding the minimax solutions is generally hard. Instead, we design strategies based on other motivations, and show that they guarantee small worst-case regret. In several cases we also prove lower bounds that show that these strategies are minimax up to constant factors.

For the problems that are not considered in this dissertation, we refer the interested reader to the standard textbook [25] as a starting point.