Modern Algorithms in Machine Learning

Lecture 2: Online Learning with Experts

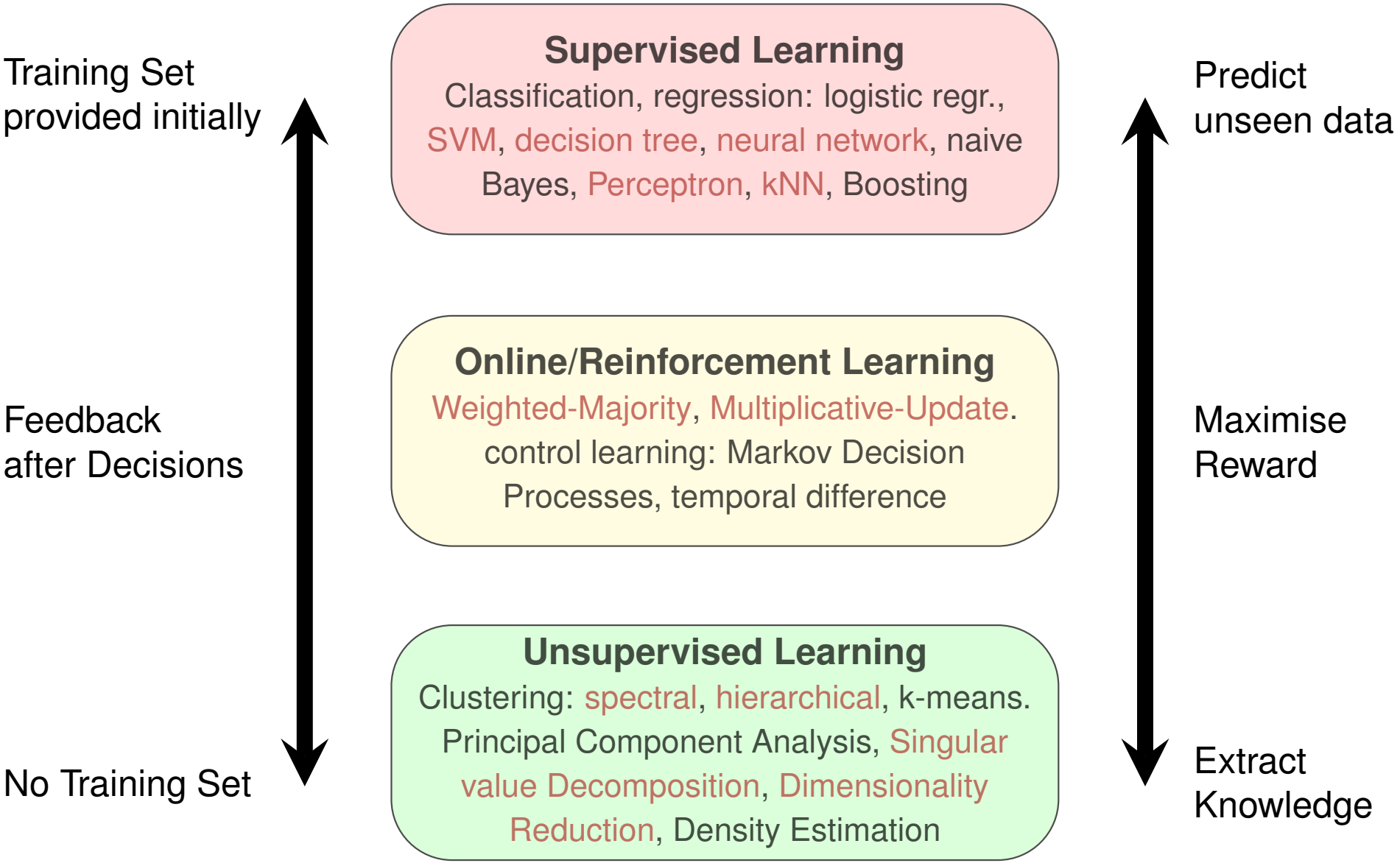
Thomas Sauerwald

University of Cambridge, Department of Computer Science and Technology
email: thomas.sauerwald@cl.cam.ac.uk

July 2020

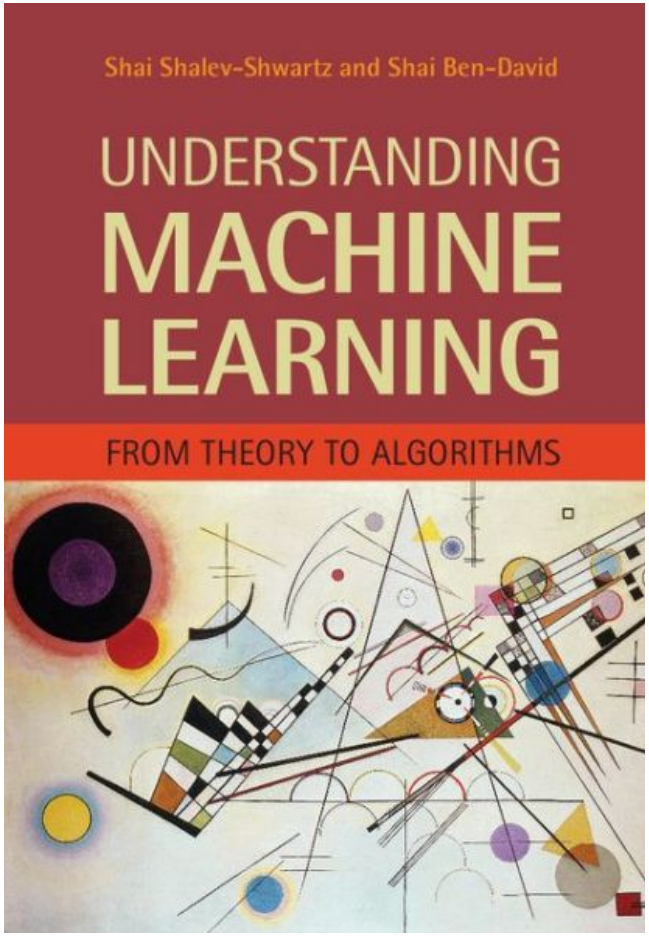
Outline

Landscape of Machine Learning Algorithms



Acknowledgments

This presentation uses some material from this textbook:



A High-Level Description of (Supervised) Machine Learning

The subject of this book is automated learning, or, as we will more often call it, Machine Learning (ML). That is, we wish to program computers so that they can "learn" from input available to them. Roughly speaking, learning is the process of converting experience into expertise or knowledge. The input to a learning algorithm is training data, representing experience, and the output is some expertise, which usually takes the form of another computer program that can perform some task. Seeking a formal-mathematical understanding of this concept, we'll have to be more explicit about what we mean by each of the involved terms: What is the training data our programs will access? How can the process of learning be automated? How can we evaluate the success of such a process (namely, the quality of the output of a learning program)?

Types of Learning and Learning Paradigms

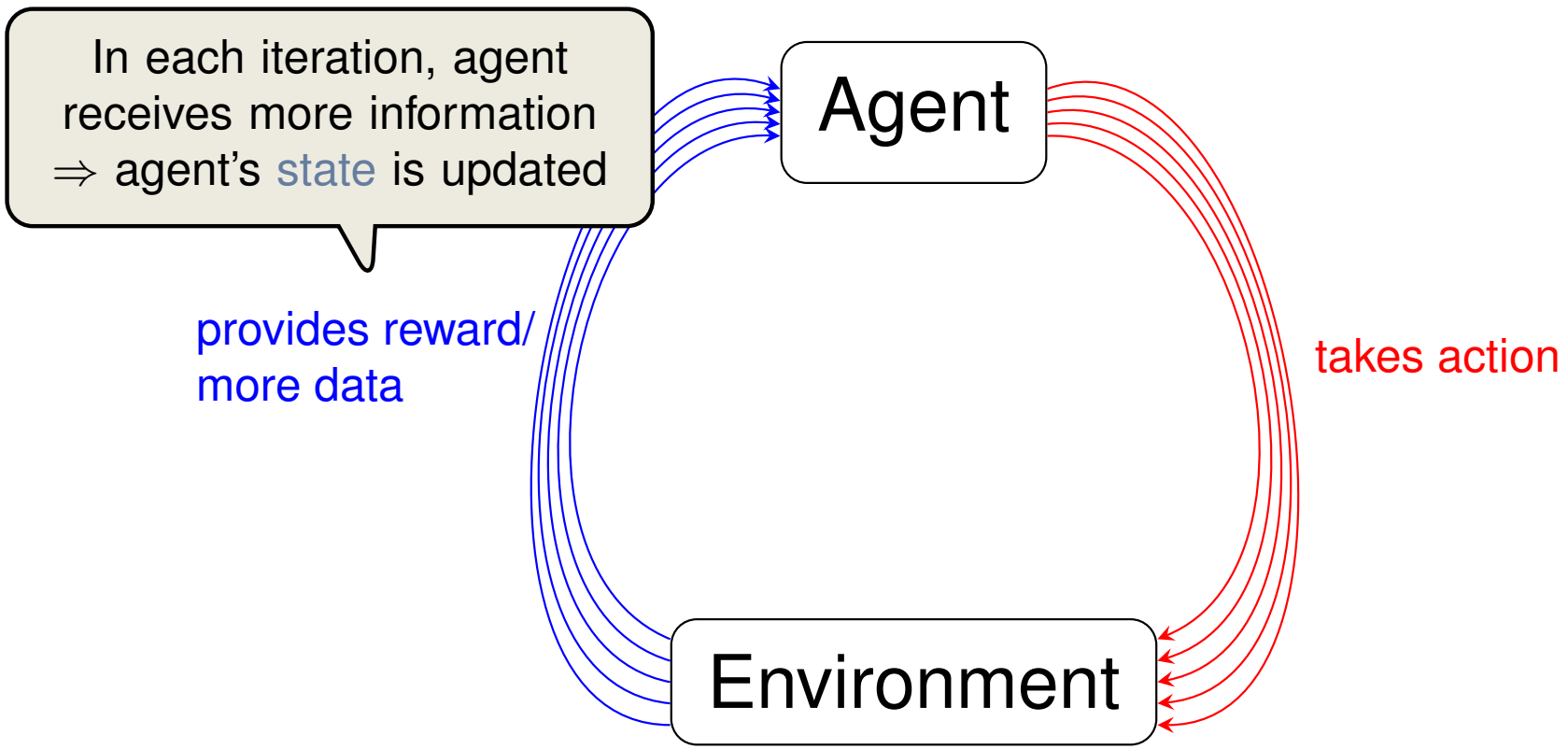
Learning: Interaction between a Learner and the environment

- Supervised versus Unsupervised.
  - Supervised Learning. Learner is provided with a large set of training examples (say, spam/not-spam emails)
  - Unsupervised Learning. There is input, but no test/training data. Goal is to extract knowledge or create a compressed version of the data.
  - Online/Reinforcement Learning. A mixed sequence of feedback and predictions (training set is revealed during the evaluation on the test set)

- Helpfulness of the Teacher.
  Human learning often involves a teacher, but when a scientist learns about nature, the environment (=teacher) is passive (assume training data is generated by a random process $\rightsquigarrow$ statistical learning)

- Online vs. Batch Learning.
  - Batch Learning. In many data mining settings, the learner has large amounts of training data at his disposal before having to make conclusions.
  - Online Learning. A stockbroker has to make a decision every day, based on the experience collected so far.

Online Algorithm/Reinforcement Learning Framework



Iteration: 5

Outline

Source: Yahoo Finance, 14 Jan 2020

Alibaba Group Holding Ltd (NYSE:BABA)

Add to Watch List | Set Alert

167.23
Delayed Data
As of Sep 30

+1.25 / +0.75%
Today's Change

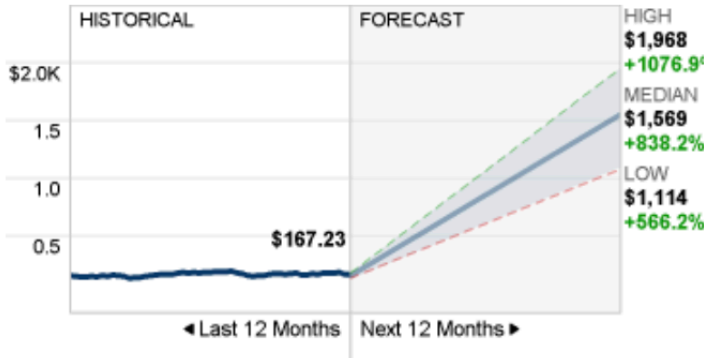
TODAY
129.77 — 195.72
52-Week Range

+22.00%
Year-to-Date

Quote | Profile | News | Charts | Forecasts | Financials | Shareholders | Competitors

Stock Price Forecast

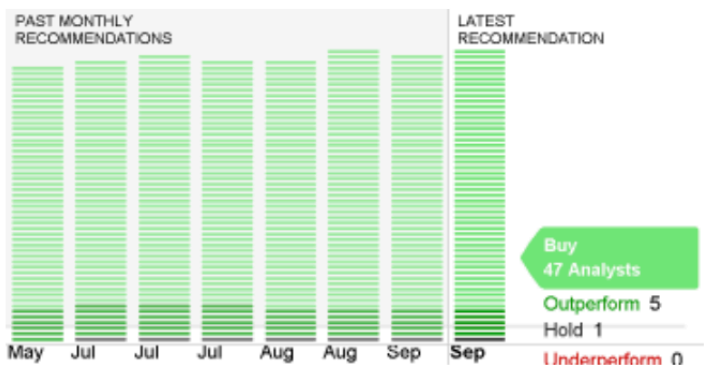
The 51 analysts offering 12-month price forecasts for Alibaba Group Holding Ltd have a median target of 1,569.17, with a high estimate of 1,968.43 and a low estimate of 1,114.28. The median estimate represents a +838.33% increase from the last price of 167.23.



Analyst Recommendations

The current consensus among 53 polled investment analysts is to **buy** stock in Alibaba Group Holding Ltd. This rating has held steady since September, when it was unchanged from a buy rating.

Move your mouse over past months for detail



Source: CNN Money, 1 Oct 2019

Online Learning using Expert Advice

Basic Setup

- Assume there is a single stock, and daily price movement is a sequence of binary events (up/down)
- The stock movements can be arbitrary (i.e., adversarial)
- We are allowed to watch n experts (these might be arbitrarily bad and correlated)

Weighted Majority Algorithm

Initialization: Fix $\delta \leq 1/2$. For every $i \in [n]$, let $w_i^{(1)} := 1$

Update: For $t = 1, 2, \ldots, T$:

- Make prediction which is the weighted majority of the experts' predictions
- For every expert i who predicts wrongly, decrease his weight by a factor of $(1 - \delta)$:

$$w_i^{(t+1)} = (1 - \delta) w_i^{(t)}$$

Example of an **ensemble method**, combining advice from several other algorithms.

Weighted Majority Algorithm: Exercise 1

Let $\delta = 1/2$, $n = 3$

t	Expert Weights	Expert Predictions	Our Pred.	Result	Our Errors
1	1, 1, 1	1, 1, 0	?	1	?
2	?, ?, ?	0, 1, 0	?	1	?
3	?, ?, ?	1, 0, 1	?	0	?
4	?, ?, ?	0, 1, 1	?	0	?
5	?, ?, ?	1, 1, 0	?	1	?
6	?, ?, ?	0, 1, 1	?	1	?
7	?, ?, ?	0, 1, 0	?	0	?
8	?, ?, ?	1, 0, 1	?	1	?
9	?, ?, ?	0, 0, 0	?	0	?
10	?, ?, ?	1, 0, 1	?	0	?
11	?, ?, ?	–	–	–	–

Weighted Majority Algorithm: Exercise 1 (Solution)

Let $\delta = 1/2$, $n = 3$

t	Expert Weights	Expert Predictions	Our Pred.	Result	Our Errors
1	$1, 1, 1$	$1, 1, 0$	1 ✓	1	0
2	$1, 1, 1/2$	$0, 1, 0$	0 ✗	1	1
3	$1/2, 1, 1/4$	$1, 0, 1$	0 ✓	0	1
4	$1/4, 1, 1/8$	$0, 1, 1$	1 ✗	0	2
5	$1/4, 1/2, 1/16$	$1, 1, 0$	1 ✓	1	2
6	$1/4, 1/2, 1/32$	$0, 1, 1$	1 ✓	1	2
7	$1/8, 1/2, 1/32$	$0, 1, 0$	1 ✗	0	3
8	$1/8, 1/4, 1/32$	$1, 0, 1$	0 ✗	1	4
9	$1/8, 1/8, 1/32$	$0, 0, 0$	0 ✓	0	4
10	$1/8, 1/8, 1/32$	$1, 0, 1$	1 ✗	0	5
11	$1/16, 1/8, 1/64$	–	–	–	–

$\Rightarrow$ We made 5 mistakes, while the best expert made only 3 mistakes.
This looks quite bad, but the example is **too small** to draw conclusions!

Analysis of the Weighted Majority Algorithm

Notation: Let $m_i^{(t)}$ be the number of mistakes of expert i after t steps.

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies

$$M^{(T)} \leq 2(1+\delta) \cdot \min_{i \in [n]} m_i^{(T)} + \frac{2 \ln n}{\delta}.$$

Proof:

- By induction, $w_i^{(t+1)} = (1-\delta)^{m_i^{(t)}}$ (see example!)
- Define a potential function $\Phi^{(t)} = \sum_{i=1}^{n} w_i^{(t)}$, so that $\Phi^{(1)} = n$.
- Every time we are wrong, also the weighted majority of experts is wrong $\Rightarrow$ at least half the total weight decreases by $1-\delta$:

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (1-\delta) \right) = \Phi^{(t)} \cdot (1 - \delta/2) \,.$$

- Hence by induction, $\Phi^{(T+1)} \leq n \cdot (1-\delta/2)^{M^{(T)}}$, but also $\Phi^{(T+1)} \geq w_i^{(T+1)}$.
- Taking logs:

$$m_i^{(T)} \ln(1-\delta) \leq M^{(T)} \ln(1-\delta/2) + \ln(n).$$

- Using now that $-\delta \geq \ln(1-\delta) \geq -\delta - \delta^2$ completes the proof. $\square$

Exercise Question 2

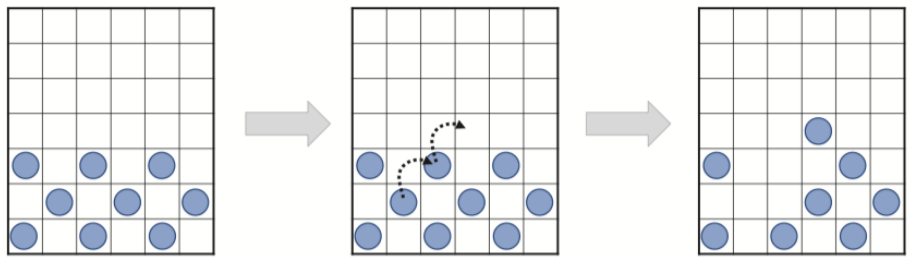
Is the following inequality always true???

$$M^{(T)} \geq \min_{i \in [n]} m_i^{(T)}$$

That means, does our algorithm always make at least as many mistakes as the best expert?

Exercise Question 3: How to Design Potential Functions

Question 3 (TS)*. In the game of solitaire chequers, the goal is to move one piece to the top row, via moves of the type shown. A move consists of a diagonal jump by one piece over another; the latter piece is then removed.



Define a potential function

$$\Phi(\text{board state}) = \sum_{\text{pieces } p} \phi\big(y\text{-coordinate of } p\big),$$

where you should define ϕ in such a way that any valid move leaves Φ unchanged. Use this potential function to prove that it is impossible to win the game, starting from the board configuration on the left.

Simulation of the Deterministic Weighted Majority Algorithm (1/2)

~/Desktop — nano weight.cpp ~/Desktop — -bash

```
Thomass-MacBook-Pro-2:Desktop thomassauerwald$ ./a.out
 ** Run of the (Deterministic) Weighted Majority Algorithm **
 Number of Experts: 6
 Probability of Mistake by Expert 0: 0.9
 Probability of Mistake by Expert 1: 0.8
 Probability of Mistake by Expert 2: 0.7
 Probability of Mistake by Expert 3: 0.5
 Probability of Mistake by Expert 4: 0.38
 Probability of Mistake by Expert 5: 0.35
Learning Rate: 0 Steps: 1000 Weight of Best Expert: 0.166667 Mistakes by Best Expert: 360 Our Mistakes: 550
Learning Rate: 0.01 Steps: 1000 Weight of Best Expert: 0.462807 Mistakes by Best Expert: 360 Our Mistakes: 444
Learning Rate: 0.02 Steps: 1000 Weight of Best Expert: 0.548359 Mistakes by Best Expert: 360 Our Mistakes: 400
Learning Rate: 0.03 Steps: 1000 Weight of Best Expert: 0.593425 Mistakes by Best Expert: 360 Our Mistakes: 386
Learning Rate: 0.04 Steps: 1000 Weight of Best Expert: 0.628579 Mistakes by Best Expert: 360 Our Mistakes: 385
Learning Rate: 0.05 Steps: 1000 Weight of Best Expert: 0.660532 Mistakes by Best Expert: 360 Our Mistakes: 380
Learning Rate: 0.06 Steps: 1000 Weight of Best Expert: 0.69085 Mistakes by Best Expert: 360 Our Mistakes: 384
Learning Rate: 0.07 Steps: 1000 Weight of Best Expert: 0.719776 Mistakes by Best Expert: 360 Our Mistakes: 387
Learning Rate: 0.08 Steps: 1000 Weight of Best Expert: 0.74724 Mistakes by Best Expert: 360 Our Mistakes: 379
Learning Rate: 0.09 Steps: 1000 Weight of Best Expert: 0.773124 Mistakes by Best Expert: 360 Our Mistakes: 375
Learning Rate: 0.1 Steps: 1000 Weight of Best Expert: 0.797329 Mistakes by Best Expert: 360 Our Mistakes: 373
Learning Rate: 0.11 Steps: 1000 Weight of Best Expert: 0.819792 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.12 Steps: 1000 Weight of Best Expert: 0.840484 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.13 Steps: 1000 Weight of Best Expert: 0.859411 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.14 Steps: 1000 Weight of Best Expert: 0.876608 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.15 Steps: 1000 Weight of Best Expert: 0.892136 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.16 Steps: 1000 Weight of Best Expert: 0.906072 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.17 Steps: 1000 Weight of Best Expert: 0.918511 Mistakes by Best Expert: 360 Our Mistakes: 371
Learning Rate: 0.18 Steps: 1000 Weight of Best Expert: 0.929554 Mistakes by Best Expert: 360 Our Mistakes: 370
Learning Rate: 0.19 Steps: 1000 Weight of Best Expert: 0.93931 Mistakes by Best Expert: 360 Our Mistakes: 370
Learning Rate: 0.2 Steps: 1000 Weight of Best Expert: 0.947889 Mistakes by Best Expert: 360 Our Mistakes: 370
Learning Rate: 0.21 Steps: 1000 Weight of Best Expert: 0.9554 Mistakes by Best Expert: 360 Our Mistakes: 370
Learning Rate: 0.22 Steps: 1000 Weight of Best Expert: 0.961948 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.23 Steps: 1000 Weight of Best Expert: 0.967634 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.24 Steps: 1000 Weight of Best Expert: 0.972553 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.25 Steps: 1000 Weight of Best Expert: 0.976794 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.26 Steps: 1000 Weight of Best Expert: 0.980437 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.27 Steps: 1000 Weight of Best Expert: 0.983556 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.28 Steps: 1000 Weight of Best Expert: 0.986219 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.29 Steps: 1000 Weight of Best Expert: 0.988483 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.3 Steps: 1000 Weight of Best Expert: 0.990404 Mistakes by Best Expert: 360 Our Mistakes: 369
Learning Rate: 0.31 Steps: 1000 Weight of Best Expert: 0.992028 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.32 Steps: 1000 Weight of Best Expert: 0.993397 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.33 Steps: 1000 Weight of Best Expert: 0.994547 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.34 Steps: 1000 Weight of Best Expert: 0.995511 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.35 Steps: 1000 Weight of Best Expert: 0.996316 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.36 Steps: 1000 Weight of Best Expert: 0.996987 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.37 Steps: 1000 Weight of Best Expert: 0.997543 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.38 Steps: 1000 Weight of Best Expert: 0.998004 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.39 Steps: 1000 Weight of Best Expert: 0.998383 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.4 Steps: 1000 Weight of Best Expert: 0.998696 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.41 Steps: 1000 Weight of Best Expert: 0.998951 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.42 Steps: 1000 Weight of Best Expert: 0.99916 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.43 Steps: 1000 Weight of Best Expert: 0.99933 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.44 Steps: 1000 Weight of Best Expert: 0.999468 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.45 Steps: 1000 Weight of Best Expert: 0.999579 Mistakes by Best Expert: 360 Our Mistakes: 368
Learning Rate: 0.46 Steps: 1000 Weight of Best Expert: 0.999668 Mistakes by Best Expert: 360 Our Mistakes: 367
Learning Rate: 0.47 Steps: 1000 Weight of Best Expert: 0.99974 Mistakes by Best Expert: 360 Our Mistakes: 367
Learning Rate: 0.48 Steps: 1000 Weight of Best Expert: 0.999797 Mistakes by Best Expert: 360 Our Mistakes: 367
Learning Rate: 0.49 Steps: 1000 Weight of Best Expert: 0.999842 Mistakes by Best Expert: 360 Our Mistakes: 368
Thomass-MacBook-Pro-2:Desktop thomassauerwald$
```

Simulation of the Deterministic Weighted Majority Algorithm (2/2)

~/Desktop — nano weight.cpp | ~/Desktop — -bash

```
Thomass-MacBook-Pro-2:Desktop thomassauerwald$ ./a.out
 ** Run of the (Deterministic) Weighted Majority Algorithm **
 Number of Experts: 6
 Probability of Mistake by Expert 0: 0.9
 Probability of Mistake by Expert 1: 0.8
 Probability of Mistake by Expert 2: 0.7
 Probability of Mistake by Expert 3: 0.23
 Probability of Mistake by Expert 4: 0.22
 Probability of Mistake by Expert 5: 0.21
Learning Rate: 0 Steps: 1000 Final Weight of Last Expert: 0.166667 Mistakes by Best Expert: 211 Our Mistakes: 306
Learning Rate: 0.01 Steps: 1000 Final Weight of Last Expert: 0.323929 Mistakes by Best Expert: 211 Our Mistakes: 162
Learning Rate: 0.02 Steps: 1000 Final Weight of Last Expert: 0.316468 Mistakes by Best Expert: 211 Our Mistakes: 138
Learning Rate: 0.03 Steps: 1000 Final Weight of Last Expert: 0.307098 Mistakes by Best Expert: 211 Our Mistakes: 136
Learning Rate: 0.04 Steps: 1000 Final Weight of Last Expert: 0.297126 Mistakes by Best Expert: 211 Our Mistakes: 136
Learning Rate: 0.05 Steps: 1000 Final Weight of Last Expert: 0.286601 Mistakes by Best Expert: 211 Our Mistakes: 133
Learning Rate: 0.06 Steps: 1000 Final Weight of Last Expert: 0.275572 Mistakes by Best Expert: 211 Our Mistakes: 136
Learning Rate: 0.07 Steps: 1000 Final Weight of Last Expert: 0.264099 Mistakes by Best Expert: 211 Our Mistakes: 144
Learning Rate: 0.08 Steps: 1000 Final Weight of Last Expert: 0.25225 Mistakes by Best Expert: 211 Our Mistakes: 154
Learning Rate: 0.09 Steps: 1000 Final Weight of Last Expert: 0.2401 Mistakes by Best Expert: 211 Our Mistakes: 157
Learning Rate: 0.1 Steps: 1000 Final Weight of Last Expert: 0.227729 Mistakes by Best Expert: 211 Our Mistakes: 160
Learning Rate: 0.11 Steps: 1000 Final Weight of Last Expert: 0.215223 Mistakes by Best Expert: 211 Our Mistakes: 166
Learning Rate: 0.12 Steps: 1000 Final Weight of Last Expert: 0.202668 Mistakes by Best Expert: 211 Our Mistakes: 169
Learning Rate: 0.13 Steps: 1000 Final Weight of Last Expert: 0.190151 Mistakes by Best Expert: 211 Our Mistakes: 172
Learning Rate: 0.14 Steps: 1000 Final Weight of Last Expert: 0.177756 Mistakes by Best Expert: 211 Our Mistakes: 174
Learning Rate: 0.15 Steps: 1000 Final Weight of Last Expert: 0.165564 Mistakes by Best Expert: 211 Our Mistakes: 175
Learning Rate: 0.16 Steps: 1000 Final Weight of Last Expert: 0.15365 Mistakes by Best Expert: 211 Our Mistakes: 177
Learning Rate: 0.17 Steps: 1000 Final Weight of Last Expert: 0.142082 Mistakes by Best Expert: 211 Our Mistakes: 179
Learning Rate: 0.18 Steps: 1000 Final Weight of Last Expert: 0.130919 Mistakes by Best Expert: 211 Our Mistakes: 179
Learning Rate: 0.19 Steps: 1000 Final Weight of Last Expert: 0.120213 Mistakes by Best Expert: 211 Our Mistakes: 184
Learning Rate: 0.2 Steps: 1000 Final Weight of Last Expert: 0.110004 Mistakes by Best Expert: 211 Our Mistakes: 183
Learning Rate: 0.21 Steps: 1000 Final Weight of Last Expert: 0.100324 Mistakes by Best Expert: 211 Our Mistakes: 189
Learning Rate: 0.22 Steps: 1000 Final Weight of Last Expert: 0.0911938 Mistakes by Best Expert: 211 Our Mistakes: 191
Learning Rate: 0.23 Steps: 1000 Final Weight of Last Expert: 0.082628 Mistakes by Best Expert: 211 Our Mistakes: 193
Learning Rate: 0.24 Steps: 1000 Final Weight of Last Expert: 0.0746311 Mistakes by Best Expert: 211 Our Mistakes: 193
Learning Rate: 0.25 Steps: 1000 Final Weight of Last Expert: 0.0672007 Mistakes by Best Expert: 211 Our Mistakes: 198
Learning Rate: 0.26 Steps: 1000 Final Weight of Last Expert: 0.060328 Mistakes by Best Expert: 211 Our Mistakes: 198
Learning Rate: 0.27 Steps: 1000 Final Weight of Last Expert: 0.0539985 Mistakes by Best Expert: 211 Our Mistakes: 198
Learning Rate: 0.28 Steps: 1000 Final Weight of Last Expert: 0.0481936 Mistakes by Best Expert: 211 Our Mistakes: 200
Learning Rate: 0.29 Steps: 1000 Final Weight of Last Expert: 0.0428907 Mistakes by Best Expert: 211 Our Mistakes: 199
Learning Rate: 0.3 Steps: 1000 Final Weight of Last Expert: 0.0380649 Mistakes by Best Expert: 211 Our Mistakes: 204
Learning Rate: 0.31 Steps: 1000 Final Weight of Last Expert: 0.0336891 Mistakes by Best Expert: 211 Our Mistakes: 204
Learning Rate: 0.32 Steps: 1000 Final Weight of Last Expert: 0.0297353 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.33 Steps: 1000 Final Weight of Last Expert: 0.0261747 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.34 Steps: 1000 Final Weight of Last Expert: 0.0229788 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.35 Steps: 1000 Final Weight of Last Expert: 0.0201191 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.36 Steps: 1000 Final Weight of Last Expert: 0.0175683 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.37 Steps: 1000 Final Weight of Last Expert: 0.0152997 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.38 Steps: 1000 Final Weight of Last Expert: 0.0132881 Mistakes by Best Expert: 211 Our Mistakes: 207
Learning Rate: 0.39 Steps: 1000 Final Weight of Last Expert: 0.0115095 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.4 Steps: 1000 Final Weight of Last Expert: 0.00994144 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.41 Steps: 1000 Final Weight of Last Expert: 0.00856299 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.42 Steps: 1000 Final Weight of Last Expert: 0.00735465 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.43 Steps: 1000 Final Weight of Last Expert: 0.00629846 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.44 Steps: 1000 Final Weight of Last Expert: 0.00537791 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.45 Steps: 1000 Final Weight of Last Expert: 0.00457791 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.46 Steps: 1000 Final Weight of Last Expert: 0.00388472 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.47 Steps: 1000 Final Weight of Last Expert: 0.00328588 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.48 Steps: 1000 Final Weight of Last Expert: 0.00277012 Mistakes by Best Expert: 211 Our Mistakes: 216
Learning Rate: 0.49 Steps: 1000 Final Weight of Last Expert: 0.00232732 Mistakes by Best Expert: 211 Our Mistakes: 216
Thomass-MacBook-Pro-2:Desktop thomassauerwald$
```

Exercise 4: Beating Weighted Majority in this Setting

- Assume we know (or learn!) that each expert i predicts each time wrongly with probability p_i

  (and this probability does not depend on whether the outcome is 1 or 0)
- How would you make your prediction?
  Hint: Use a maximum likelihood approach ($\leadsto$ Naive Bayes)

Improving the Weighted Majority Algorithm?

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$M^{(T)} \leq 2(1+\delta) \cdot m_i^{(T)} + \frac{2 \ln n}{\delta}.$$

Question: Is there a way to avoid the factor of 2?

Exercise 5: For any **deterministic** algorithm, the factor of 2 cannot be avoided!

Idea: Employ a randomised strategy which selects an expert with probability proportional to its success!

Randomised Weighted Majority

Randomised Weighted Majority Algorithm

Initialization: Fix $\delta \leq 1/2$. For every $i \in [n]$, let $w_i^{(1)} := 1$

Update: For $t = 1, 2, \ldots, T$:

- Pick expert i with probability proportional to w_i and follow that prediction
- For every expert i who predicts wrongly, decrease his weight by a factor of $(1 - \delta)$:

$$w_i^{(t+1)} = (1 - \delta) w_i^{(t)}$$

Note that the number of mistakes we are making is now a random variable!

Exercise Question 6

Consider the following run of the Deterministic Weighted Majority Algorithm:

t	Weights	Predictions	Actual Result	Our Prediction	Our Errors
1	1,1	1,0	0	1	1
2	1/2,1	1,0	1	0	2
3	1/2,1/2	0,1	1	0	3
4	1/4,1/2	1,0	1	0	4
5	1/4,1/4	–	–	–	–

Consider now the Randomised Weighted Majority Algorithm and compute the expected number of mistakes ($\mathbf{E}\left[M^{(4)}\right]$)

Hint: Running a randomised algorithm can be visualised by a path from the root of a tree to a leaf, where the directions of the path depend on the random choices.

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^{n} w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

- Similar to before, the potential change (which is deterministic!) satisfies

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(1 - \delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right) \leq \Phi^{(t)} \exp\left(-\delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$$

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

- Similar to before, the potential change (which is deterministic!) satisfies

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(1 - \delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right) \leq \Phi^{(t)} \exp\left(-\delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$$

- Thus the final potential is at most $\Phi^{(0)} \exp\left(-\delta \sum_{t=1}^{T} \lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$, and

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

- Similar to before, the potential change (which is deterministic!) satisfies

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(1 - \delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right) \leq \Phi^{(t)} \exp\left(-\delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$$

- Thus the final potential is at most $\Phi^{(0)} \exp\left(-\delta \sum_{t=1}^T \lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$, and

$$\Phi^{(t+1)} \geq \prod_{t=1}^{T} \left(1 - \delta\lambda_i^{(t)}\right) \geq (1-\delta)^{\sum_{t=1}^T \lambda_i^{(t)}}$$

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

- Similar to before, the potential change (which is deterministic!) satisfies

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(1 - \delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right) \leq \Phi^{(t)} \exp\left(-\delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$$

- Thus the final potential is at most $\Phi^{(0)} \exp\left(-\delta \sum_{t=1}^T \lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$, and

$\geq$ Final weight of expert i

$$\Phi^{(t+1)} \geq \prod_{t=1}^{T} \left(1 - \delta\lambda_i^{(t)}\right) \geq (1-\delta)^{\sum_{t=1}^T \lambda_i^{(t)}}$$

Analysis of Randomised Weighted Majority

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

This was a factor of 2 before!

Proof Sketch:

- The probability of picking expert i in round t is $p_i^{(t)} := \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}} = \frac{w_i^{(t)}}{\Phi^{(t)}}$.
- Let $\lambda_i^{(t)}$ be 1 iff expert i is wrong at time t (and 0 otherwise)
- Then the expected number of mistakes by our algorithm is

$$\sum_{t=0}^{T-1} \lambda^{(t)} \cdot \mathbf{p}^{(t)}$$

- Similar to before, the potential change (which is deterministic!) satisfies

$$\Phi^{(t+1)} \leq \Phi^{(t)} \cdot \left(1 - \delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right) \leq \Phi^{(t)} \exp\left(-\delta\lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$$

- Thus the final potential is at most $\Phi^{(0)} \exp\left(-\delta \sum_{t=1}^T \lambda^{(t)} \cdot \mathbf{p}^{(t)}\right)$, and

$\geq$ Final weight of expert i

$$\Phi^{(t+1)} \geq \prod_{t=1}^{T} \left(1 - \delta\lambda_i^{(t)}\right) \geq (1-\delta)^{\sum_{t=1}^T \lambda_i^{(t)}}$$

Final step of the proof is similar: Take logs etc.

Exercise 7: Dealing with poor experts

- Suppose there might be some experts who make the wrong prediction more often than the correct one (however, we don't know the identity of these experts).
- Can we modify the algorithm to do well also in this case?

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2 \ln n}{\delta}.$$

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2 \ln n}{\delta}.$$

Interpretation:

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2 \ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance
- Picking learning rate $\delta = \sqrt{2\ln(n)/T}$
  (assuming T is large enough so that $\delta \leq 1/2$!)

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance
- Picking learning rate $\delta = \sqrt{2\ln(n)/T}$
  (assuming T is large enough so that $\delta \leq 1/2$!)

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{2\ln(n)/T} \cdot T + \sqrt{2\ln(n) \cdot T}$$

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance
- Picking learning rate $\delta = \sqrt{2\ln(n)/T}$
  (assuming T is large enough so that $\delta \leq 1/2$!)

$$\begin{aligned}\mathbf{E}\left[M^{(T)}\right] &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{2\ln(n)/T} \cdot T + \sqrt{2\ln(n)\cdot T} \\ &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}\end{aligned}$$

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance
- Picking learning rate $\delta = \sqrt{2\ln(n)/T}$
  (assuming T is large enough so that $\delta \leq 1/2$!)

$$\begin{aligned}\mathbf{E}\left[M^{(T)}\right] &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{2\ln(n)/T} \cdot T + \sqrt{2\ln(n)\cdot T} \\ &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}\end{aligned}$$

Additive error negligible in most cases compared to $\min_{i\in[n]} m_i^{(T)}$!

Impact of the Learning Rate

Analysis

The number of mistakes of our algorithm $M^{(T)}$ satisfies for any expert i,

$$\mathbf{E}\left[M^{(T)}\right] \leq 1 \cdot (1+\delta) \cdot m_i^{(T)} + \frac{2\ln n}{\delta}.$$

Interpretation:

- Suppose that T is known in advance
- Picking learning rate $\delta = \sqrt{2\ln(n)/T}$
  (assuming T is large enough so that $\delta \leq 1/2$!)

$$\begin{aligned}\mathbf{E}\left[M^{(T)}\right] &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{2\ln(n)/T} \cdot T + \sqrt{2\ln(n)\cdot T} \\ &\leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}\end{aligned}$$

Additive error negligible in most cases compared to $\min_{i\in[n]} m_i^{(T)}$!

Can we do better than that?

A “Pathological” Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?
  - Every expert $i \in [n]$ will make $T/2 \pm \Theta(\sqrt{T})$ many mistakes

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?
  - Every expert $i \in [n]$ will make $T/2 \pm \Theta(\sqrt{T})$ many mistakes
  - Best expert will make $T/2 - \Theta(\sqrt{T\ln(n)})$ many mistakes (proof omitted)

A "Pathological" Instance

Corollary

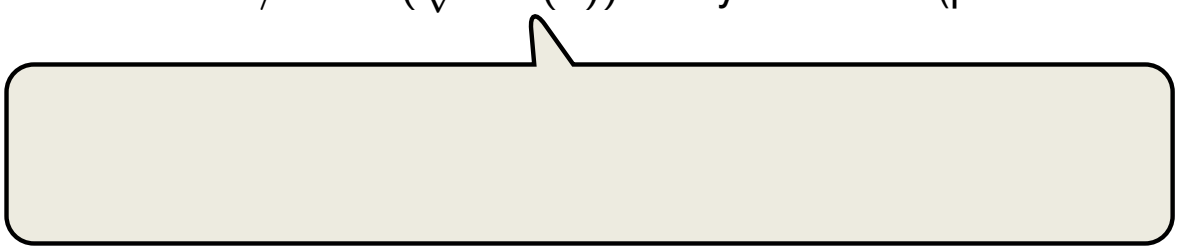
For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?
  - Every expert $i \in [n]$ will make $T/2 \pm \Theta(\sqrt{T})$ many mistakes
  - Best expert will make $T/2 - \Theta(\sqrt{T\ln(n)})$ many mistakes (proof omitted)



A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?
  - Every expert $i \in [n]$ will make $T/2 \pm \Theta(\sqrt{T})$ many mistakes
  - Best expert will make $T/2 - \Theta(\sqrt{T\ln(n)})$ many mistakes (proof omitted)

  - This demonstrates tightness of the error term

A "Pathological" Instance

Corollary

For $\delta = \sqrt{2\ln(n)/T}$, the number of our mistakes $M^{(T)}$ satisfies

$$\mathbf{E}\left[M^{(T)}\right] \leq \min_{i\in[n]} m_i^{(T)} + \sqrt{8T\ln(n)}.$$

- Suppose every expert $i = 1, 2, \ldots, n$ flips an unbiased coin, and the result is also an unbiased coin flip (independent of the experts' predictions)
- $\Rightarrow$ Regardless of our algorithm, the number of our mistakes satisfies

$$\mathbf{E}\left[M^{(T)}\right] = T \cdot \frac{1}{2}$$

- How good is the best expert?
  - Every expert $i \in [n]$ will make $T/2 \pm \Theta(\sqrt{T})$ many mistakes
  - Best expert will make $T/2 - \Theta(\sqrt{T\ln(n)})$ many mistakes (proof omitted)

  - This demonstrates tightness of the error term
  - Best expert will be good just by chance!

Exercise Question 8

We have seen a good choice for the learning if T is known in advance, but what can we do if T is not known?

Exercise Question 8

We have seen a good choice for the learning if T is known in advance, but what can we do if T is not known?

Hint: Instead of a fixed learning rate, decrease learning rate over time.

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

Coming back to our example of **stock prediction**:

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

Coming back to our example of **stock prediction**:

- could define cost $m_j^{(t)} = 0$ if expert j is neutral (HOLD)

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

Coming back to our example of **stock prediction**:

- could define cost $m_j^{(t)} = 0$ if expert j is neutral (HOLD)
- cost $m_j^{(t)} > 0$ if expert j makes the wrong prediction
  (closer to 1 the stronger prediction and stronger the price change)

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

Coming back to our example of **stock prediction**:

- could define cost $m_j^{(t)} = 0$ if expert j is neutral (HOLD)
- cost $m_j^{(t)} > 0$ if expert j makes the wrong prediction
  (closer to 1 the stronger prediction and stronger the price change)
- cost $m_j^{(t)} < 0$ if expert j makes the correct prediction

A More General Setting

New Setup

- At each step, we pick one expert i randomly out of n experts
- That expert i and our algorithm incur a cost $m_i^{(t)}$, but we also observe the costs of all experts (a vector $(m_j^{(t)})_{i=1}^n$)
- costs $m_j^{(t)}$ can be arbitrary in the range $[-1, 1]$

Coming back to our example of **stock prediction**:

- could define cost $m_j^{(t)} = 0$ if expert j is neutral (HOLD)
- cost $m_j^{(t)} > 0$ if expert j makes the wrong prediction
  (closer to 1 the stronger prediction and stronger the price change)
- cost $m_j^{(t)} < 0$ if expert j makes the correct prediction

Idea of the "Multiplicative Weights-Algorithm"

- In the first iteration, simply pick a decision uniformly at random
- Every decision will be penalised or rewarded through a multiplicative weight-update

The Multiplicative Weights Algorithm

The Multiplicative Weights Algorithm

Initialization: Fix $\delta \leq 1/2$. For every $i \in [n]$, let $w_i^{(1)} := 1$

Update: For $t = 1, 2, \ldots, T$:

- Choose expert i with prop. proportional to $w_i^{(t)}$.
- Observe the costs of all n experts in round t, $m^{(t)}$
- For every expert i, update its weight by:

$$w_i^{(t+1)} = (1 - \delta m_i^{(t)}) w_i^{(t)}$$

The Multiplicative Weights Algorithm

The Multiplicative Weights Algorithm

Initialization: Fix $\delta \leq 1/2$. For every $i \in [n]$, let $w_i^{(1)} := 1$

Update: For $t = 1, 2, \ldots, T$:

- Choose expert i with prop. proportional to $w_i^{(t)}$.
- Observe the costs of all n experts in round t, $m^{(t)}$
- For every expert i, update its weight by:

$$w_i^{(t+1)} = (1 - \delta m_i^{(t)}) w_i^{(t)}$$

Analysis

For any expert i, the expected cost of this algorithm is at most

$$\sum_{t=1}^{T} m_i^{(t)} + \delta \cdot \sum_{t=1}^{T} \left| m_i^{(t)} \right| + \frac{\log n}{\delta}.$$

The Multiplicative Weights Algorithm

The Multiplicative Weights Algorithm

Initialization: Fix $\delta \leq 1/2$. For every $i \in [n]$, let $w_i^{(1)} := 1$

Update: For $t = 1, 2, \ldots, T$:

- Choose expert i with prop. proportional to $w_i^{(t)}$.
- Observe the costs of all n experts in round t, $m^{(t)}$
- For every expert i, update its weight by:

$$w_i^{(t+1)} = (1 - \delta m_i^{(t)}) w_i^{(t)}$$

Analysis

For any expert i, the expected cost of this algorithm is at most

$$\sum_{t=1}^{T} m_i^{(t)} + \delta \cdot \sum_{t=1}^{T} \left| m_i^{(t)} \right| + \frac{\log n}{\delta}.$$

Derivation is very similar to the ones shown before.

Conclusions

Summary

- Weighted Majority Algorithm
  - natural, simple (and deterministic) algorithm
  - good performance, but could be a factor of 2 worse than the best expert
- Randomised Weighted Majority Algorithm
  - Randomised extension
  - almost optimal performance thanks to randomisation which guards against tailored worst-case instances (cmp. Quick-Sort!)
  - impact of the learning rate: small learning rate gives very good performance guarantees. However, actual performance may depend on the specific data set at hand (cf. simulations!)
- Multiplicative Weight-Update Algorithm
  - further generalisation of the (randomised) weighted majority algorithm

Conclusions

Summary

- Weighted Majority Algorithm
  - natural, simple (and deterministic) algorithm
  - good performance, but could be a factor of 2 worse than the best expert
- Randomised Weighted Majority Algorithm
  - Randomised extension
  - almost optimal performance thanks to randomisation which guards against tailored worst-case instances (cmp. Quick-Sort!)
  - impact of the learning rate: small learning rate gives very good performance guarantees. However, actual performance may depend on the specific data set at hand (cf. simulations!)
- Multiplicative Weight-Update Algorithm
  - further generalisation of the (randomised) weighted majority algorithm

Outlook

- These algorithms are examples of the Ensemble-Method: Framework combining weak predictions into a strong learner
- Similar examples will be Perceptron and AdaBoost (coming soon!)

References


S. Arora, E. Hazan and S. Kale
The Multiplicative Weights Update Method: A Meta-Algorithm and Applications
Theory of Computing, Volume 8 (2012).


A. Blum
Empirical Support for Winnow and Weighted-Majority Algorithms: Results on a Calendar Scheduling Domain
Machine Learning, Volume 26, Issue 1, pages 5-23, 1997.


A. Blum, J. Hopcroft and R. Kannan
Foundations of Data Science
https://www.cs.cornell.edu/jeh/book.pdf


N. Littlestone and M.K. Warmuth
The Weighted Majority Algorithm
Information and Computation, Volume 108, Issue 2, 1994.


S. Shalev-Shwartz and S. Ben-David
Understanding Machine Learning: From Theory to Algorithms
Cambridge University Press, 2014.
https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf