Lecture 15: Online Learning using Expert Advice

John Sylvester Nicolás Rivera Luca Zanetti Thomas Sauerwald

Lent 2020

Outline

Introduction

Online Learning with Experts

Landscape of Machine Learning Algorithms

Training Set provided initially | **Supervised Learning** | Predict unseen data

Classification, regression: logistic regr., SVM, decision tree, neural networks, naive Bayes, Perceptron, kNN, Boosting

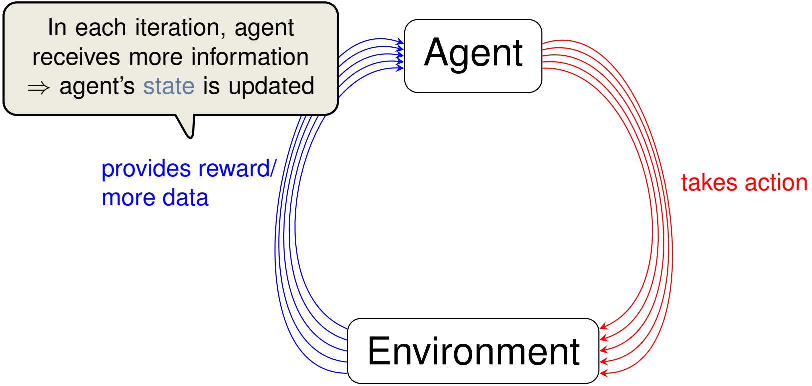
Feedback after Decisions | **Online/Reinforcement Learning** | Maximise Reward

Weighted-Majority, Multiplicative-Update. control learning: Markov Decision Processes, temporal difference

No Training Set | **Unsupervised Learning** | Extract Knowledge

Clustering: spectral, hierarchical, k-means; Dimensionality Reduction, PCA, SVD

Online Algorithm/Reinforcement Learning Framework

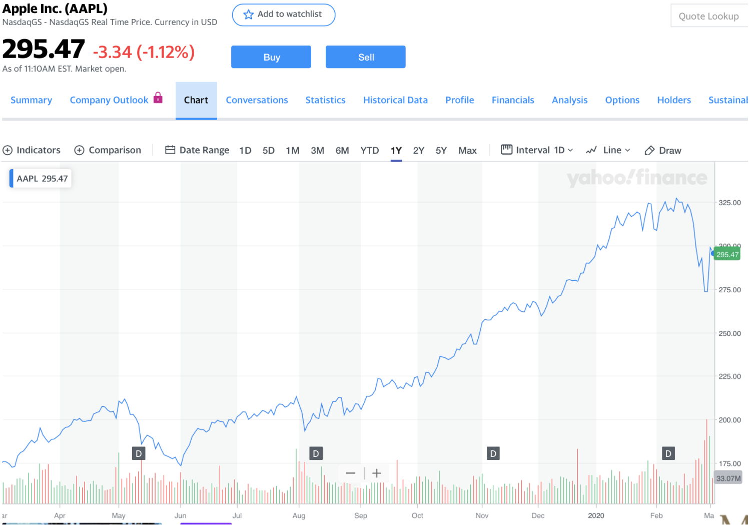


Iteration:

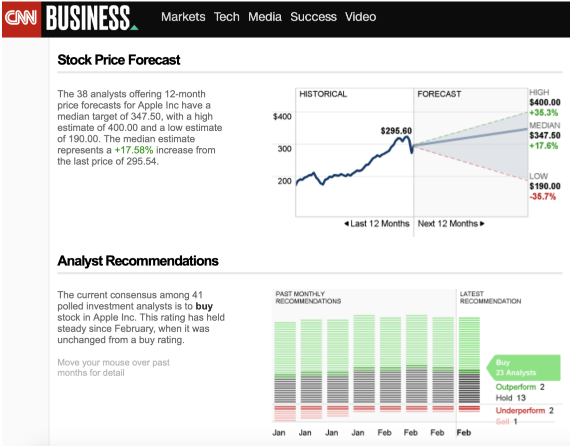
Outline

Introduction

Online Learning with Experts



Source: Yahoo Finance, 3 March 2020



Source: CNN Money, 3 March 2020

Online Learning using Expert Advice

Basic Setup

- Assume there is a single stock, and daily price movement is a sequence of binary events (up $= 1$ /down $= 0$)
- 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: Example

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	$1/16, 1/8, 1/64$	–	–	–	–

⇒ 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

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

This bound holds for any input, any T and any δ!

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 (Problem Sheet 6, Q5.2)

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?

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$
```

Improving the Weighted Majority Algorithm?

Analysis

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

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

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

Question 5.2 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!

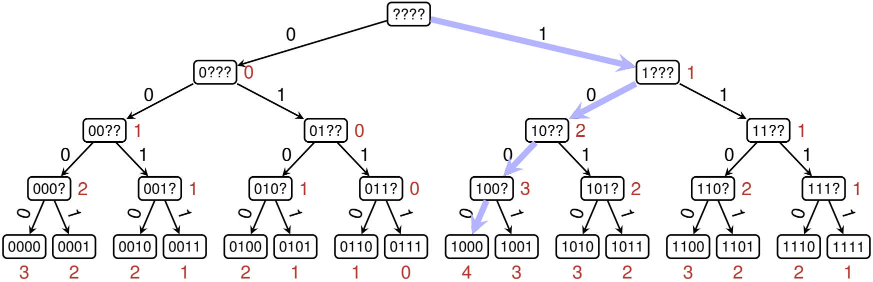
Example: Deterministic vs. Randomised Weighted Majority (1/2)

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 let us compute the expected number of mistakes ($\mathbf{E}\left[M^{(4)} \right]$)

Example: Deterministic vs. Randomised Weighted Majority (2/2)



- Let $x^{(t)}$ be a 0/1 random variable, indicating if our t-th prediction is wrong.
- Then:

$$\mathbf{E}\left[\, x^{(1)} \,\right] = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{1}{2}.$$

- Similarly, $\mathbf{E}\left[\, x^{(2)} \,\right] = \frac{2}{3}$, $\mathbf{E}\left[\, x^{(3)} \,\right] = \frac{1}{2}$ and $\mathbf{E}\left[\, x^{(4)} \,\right] = \frac{2}{3}$.
- Hence,

$$\begin{aligned} \mathbf{E}\left[\, M^{(4)} \,\right] &= \mathbf{E}\left[\, x^{(1)} + x^{(2)} + x^{(3)} + x^{(4)} \,\right] \\ &= \mathbf{E}\left[\, x^{(1)} \,\right] + \mathbf{E}\left[\, x^{(2)} \,\right] + \mathbf{E}\left[\, x^{(3)} \,\right] + \mathbf{E}\left[\, x^{(4)} \,\right] = \frac{7}{3} \end{aligned}$$

Much better than the deterministic algorithm!

Analysis of Randomised Weighted Majority

Analysis

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

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

This was a factor of 2 before!

Proof:

- Define a potential function $\Phi^{(t)} = \sum_{i=1}^{n} w_i^{(t)}$, so that $\Phi^{(1)} = n$.
- The probability of picking expert i in round t is $p_i^{(t)} := w_i^{(t)} / \sum_{j=1}^{n} w_j^{(t)} = w_i^{(t)} / \Phi^{(t)}$.
- 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 $\mathbf{E}[M^{(T)}] = \sum_{t=1}^{T} \lambda^{(t)} \cdot p^{(t)}$.
- The new potential (which is deterministic!) can be upper bounded by:

$$\Phi^{(t+1)} = \sum_{i=1}^{n} w_i^{(t+1)} = \sum_{i=1}^{n} (1 - \delta \lambda_i^{(t)}) \cdot w_i^{(t+1)} = \Phi^{(t)} \cdot \left(1 - \delta \lambda^{(t)} p^{(t)}\right) \leq \Phi^{(t)} \cdot \exp\left(-\delta \lambda^{(t)} \cdot p^{(t)}\right)$$

- Thus the final potential satisfies

$$\Phi^{(T+1)} \leq \Phi^{(1)} \cdot \exp\left(-\delta \sum_{t=1}^{T} \lambda^{(t)} \cdot p^{(t)}\right) = n \cdot \exp\left(-\delta \cdot \mathbf{E}\left[\,M^{(T)}\,\right]\right),$$

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

$\ln(1-\delta) \geq -\delta - \delta^2$

$$\Rightarrow \quad \ln n - \delta \cdot \mathbf{E}[M^{(T)}] \geq \ln(1-\delta) \cdot m_i^{(T)} \quad \Rightarrow \quad \mathbf{E}[M^{(T)}] \leq \frac{(\delta + \delta^2)}{\delta} \cdot m_i^{(T)} + \frac{\ln n}{\delta} \qquad \square$$

Optimising the Learning Rate

Analysis

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

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

Interpretation:

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

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

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

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

- 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, uses central limit theorem)

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

Extension: 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?

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:

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

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

Outlook

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

References

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

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