Advanced Topics in Machine Learning

# Part II: An Introduction to Online Learning

A. LAZARIC (*INRIA-Lille*)

*DEI, Politecnico di Milano*

SequeL – INRIA Lille

April 2-15, 2012

# Outline


Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Outline

Introduction
- The Online Prediction Game
- Binary Sequence Prediction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Outline

Introduction
- The Online Prediction Game
- Binary Sequence Prediction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Online Learning

The prediction problem

- What will be the *rain precipitation* next month?

# Online Learning

The prediction problem

- What will be the *rain precipitation* next month?
- What will be the *price of this stock* tomorrow?

# Online Learning

The prediction problem

- What will be the *rain precipitation* next month?
- What will be the *price of this stock* tomorrow?
- How *many iPad* will be sold next quarter?

# Online Learning

The prediction problem

- What will be the *rain precipitation* next month?
- What will be the *price of this stock* tomorrow?
- How *many iPad* will be sold next quarter?
- How many *contacts will have this webpage* in the next hour?

# Online Learning

The prediction problem

- What will be the *rain precipitation* next month?
- What will be the *price of this stock* tomorrow?
- How *many iPad* will be sold next quarter?
- How many *contacts will have this webpage* in the next hour?
- ...

# Online Learning vs Statistical Learning

Limitations of Statistical Learning

- Reality is not *stochastic*
- Data are often arriving in a *sequence*
- *Training and testing* are rarely separated
- *Massive* datasets must be provided in a stream

# Online Learning vs Statistical Learning (cont'd)

| | SL | OL |
|---|---|---|
| *Samples* | Batch | *In a stream* |
| *Assumptions* | Stochastic model | *Individual sequence* |
| *Analysis* | Average case | *Worst case* |
| *Performance* | Excess risk | *Regret* |

# The Prediction Game

The environment

- Outcome space $\mathcal{Y}$

# The Prediction Game

The environment

- Outcome space $\mathcal{Y}$

The learner

- Decision (prediction) space $\mathcal{D}$

# The Prediction Game

The environment

- Outcome space $\mathcal{Y}$

The learner

- Decision (prediction) space $\mathcal{D}$

The performance

- Loss function $\ell(p, y)$ with $y \in \mathcal{Y}$ and $p \in \mathcal{D}$

# The Prediction Game (cont'd)

At each round $t = 1, \dots, n$

# The Prediction Game (cont'd)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses an outcome $y_t \in \mathcal{Y}$
  - The learner chooses a prediction $\hat{p}_t \in \mathcal{D}$

# The Prediction Game (cont'd)

At each round $t = 1, \dots, n$

- At the same time
  - The environment chooses an outcome $y_t \in \mathcal{Y}$
  - The learner chooses a prediction $\hat{p}_t \in \mathcal{D}$
- The learner suffers a loss $\ell(\hat{p}_t, y_t)$

# The Prediction Game (cont'd)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses an outcome $y_t \in \mathcal{Y}$
  - The learner chooses a prediction $\hat{p}_t \in \mathcal{D}$
- The learner suffers a loss $\ell(\hat{p}_t, y_t)$
- The environment reveals $y_t$

# The Prediction Game (cont'd)

At each round $t = 1, \dots, n$ (not necessarily finite time)

- At the same time
  - The environment chooses an outcome $y_t \in \mathcal{Y}$
  - The learner chooses a prediction $\hat{p}_t \in \mathcal{D}$
- The learner suffers a loss $\ell(\hat{p}_t, y_t)$
- The environment reveals $y_t$

# Outline

Introduction
- The Online Prediction Game
- Binary Sequence Prediction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# A "Gentle" Start: Binary Sequence Prediction

**Problem**: predict (online) the next bit in an arbitrary string of bits

- $\mathcal{Y} = \mathcal{D} = \{0, 1\}$
- $\ell(p, y) = \mathbb{I}\{y \neq p\}$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Doubt**: I do not know anything about where this string is coming from... and I am not an expert of strings of bits...

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Doubt**: I do not know anything about where this string is coming from... and I am not an expert of strings of bits...
**Solution**: ask to experts!

# A “Gentle” Start: Binary Sequence Prediction (cont’d)

**Doubt**: I do not know anything about where this string is coming from... and I am not an expert of strings of bits...
**Solution**: ask to experts!

- $N$ experts
- Each returning a prediction $f_{i,t} \in \mathcal{D}$ $(i = 1, \ldots, N)$

# A “Gentle” Start: Binary Sequence Prediction (cont’d)

**Simple case**: one of my experts perfectly knows the sequence

$$\exists i,\ \forall t,\ \ \ell(y_t, f_{i,t}) = 0$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Simple case**: one of my experts perfectly knows the sequence

$$\exists i, \ \forall t, \ \ \ell(y_t, f_{i,t}) = 0$$

**Simple algorithm** the *Halving* algorithm (a.k.a. "there can be only one!"):
Initialize the weights $w_{i,0} = 1$

- Collect all the experts' predictions $f_{i,t}$
- Take $\hat{p}_t = 1$ if the *majority* of experts with $w_i = 1$ suggests 1, 0 otherwise
- Observe $y_t$
- Set $w_i = 0$ for all the $f_{i,t} \neq y_t$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Question**: how many mistakes does this algorithm make?

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

- At the beginning $m = 0$ and $W_0 = N$. *[algorithm]*

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

- At the beginning $m = 0$ and $W_0 = N$. *[algorithm]*
- At each mistake, at least half of the active experts were wrong and then removed: *[algorithm]*

$$W_m \leq \frac{W_{m-1}}{2}$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

- At the beginning $m = 0$ and $W_0 = N$. *[algorithm]*
- At each mistake, at least half of the active experts were wrong and then removed: *[algorithm]*

$$W_m \leq \frac{W_{m-1}}{2}$$

- Applying the previous relationship recursively *[math]*

$$W_m \leq \frac{W_{m-1}}{2} \leq \frac{W_{m-2}}{4} \leq \ldots \leq \frac{W_0}{2^m}$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

- At the beginning $m = 0$ and $W_0 = N$. *[algorithm]*
- At each mistake, at least half of the active experts were wrong and then removed: *[algorithm]*

$$W_m \leq \frac{W_{m-1}}{2}$$

- Applying the previous relationship recursively *[math]*

$$W_m \leq \frac{W_{m-1}}{2} \leq \frac{W_{m-2}}{4} \leq \ldots \leq \frac{W_0}{2^m}$$

- According to the "simple case", after $m$ there will always at least one expert still active *[assumption]*

$$W_m \geq 1$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

Let $W_m$ be the total number of *active* experts after $m$ mistakes.

- At the beginning $m = 0$ and $W_0 = N$. *[algorithm]*
- At each mistake, at least half of the active experts were wrong and then removed: *[algorithm]*

$$W_m \leq \frac{W_{m-1}}{2}$$

- Applying the previous relationship recursively *[math]*

$$W_m \leq \frac{W_{m-1}}{2} \leq \frac{W_{m-2}}{4} \leq \ldots \leq \frac{W_0}{2^m}$$

- According to the "simple case", after $m$ there will always at least one expert still active *[assumption]*

$$W_m \geq 1$$

- Putting together *[math]*

$$\frac{W_0}{2^m} \geq 1 \Rightarrow m \leq \lfloor \log_2 N \rfloor$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Theorem**

*For* ***any*** *binary sequence $y_1, \ldots, y_t, \ldots$, we consider a halving algorithm on $N$ experts. If one experts makes no mistake over the sequence, then*

$$m \leq \lfloor \log_2 N \rfloor$$

# A "Gentle" Start: Binary Sequence Prediction (cont'd)

**Theorem**

*For* ***any*** *binary sequence* $y_1, \ldots, y_t, \ldots$*, we consider a halving algorithm on* $N$ *experts. If one experts makes no mistake over the sequence, then*

$$m \leq \lfloor \log_2 N \rfloor$$

- No stochastic assumption!
- No high–probability result!
- Finite number of mistakes for ***ANY*** possible sequence!

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

The Continuous Prediction Game

The Exponentially Weighted Average Forecaster

Parameter Tuning

Bounds for Small Losses

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

...lusions

# Outline


Introduction

Continuous Prediction with Expert Advice: the EWA
- The Continuous Prediction Game
- The Exponentially Weighted Average Forecaster
- Parameter Tuning
- Bounds for Small Losses

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

[...]lusions

# Continuous Prediction

- Outcome space $\mathcal{Y}$ is arbitrary
- Decision space $\mathcal{D}$ is a convex subset of $\mathbb{R}^s$
- Loss function $\ell(p, y)$
  - *bounded* $(\ell : \mathcal{D} \times \mathcal{Y} \to [0, 1])$
  - *convex in the first argument* ($\ell(\cdot, y)$ is convex for any $y \in \mathcal{Y}$)

# Continuous Prediction (cont'd)

- Experts $f_{1,t}, \ldots, f_{N,t}$
- The performance measure: the (expert) regret

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_{1 \leq i \leq N} \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

# Continuous Prediction (cont'd)

- Experts $f_{1,t}, \ldots, f_{N,t}$
- The performance measure: the (expert) regret

$$R_n = \underbrace{\sum_{t=1}^{n} \ell(\hat{p}_t, y_t)}_{\text{alg. cumul. loss}} - \min_{1 \le i \le N} \underbrace{\sum_{t=1}^{n} \ell(f_{i,t}, y_t)}_{\text{expert } i \text{ cumul. loss}}$$

# Continuous Prediction (cont'd)

- Experts $f_{1,t}, \dots, f_{N,t}$
- The performance measure: the (expert) regret

$$R_n = \underbrace{\sum_{t=1}^{n} \ell(\hat{p}_t, y_t)}_{\text{alg. cumul. loss}} - \underbrace{\min_{1 \le i \le N} \sum_{t=1}^{n} \ell(f_{i,t}, y_t)}_{\text{best expert in hindsight}}$$

# Continuous Prediction (cont'd)

- Expert cumulative loss on the sequence $\mathbf{y}^n = (y_1, \dots, y_n)$

$$L_{i,n}(\mathbf{y}^n) = \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

- Algorithm $\mathcal{A}$ cumulative loss

$$L_n(\mathcal{A}; \mathbf{y}^n) = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t)$$

- Regret

$$R_n = L_n(\mathcal{A}; \mathbf{y}^n) - \min_i L_{i,n}(\mathbf{y}^n)$$

# Continuous Prediction (cont'd)

- Expert cumulative loss on the sequence $\mathbf{y}^n = (y_1, \dots, y_n)$

$$L_{i,n}(\mathbf{y}^n) = \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

- Algorithm $\mathcal{A}$ cumulative loss

$$L_n(\mathcal{A}; \mathbf{y}^n) = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t)$$

- Regret

$$R_n = L_n(\mathcal{A}; \mathbf{y}^n) - \min_i L_{i,n}(\mathbf{y}^n)$$

**Objective**: find an *alg.* with ***small regret*** for ***any*** sequence $\mathbf{y}^n$

# Continuous Prediction (cont'd)

The definition of expert is so general that almost anything fits:

# Continuous Prediction (cont'd)

The definition of expert is so general that almost anything fits:

- $f_{i,t}$ can be a function of a *context* $x \Rightarrow$ *adaptive experts*

# Continuous Prediction (cont'd)

The definition of expert is so general that almost anything fits:

- $f_{i,t}$ can be a function of a *context* $x \Rightarrow$ *adaptive experts*
- $f_{i,t}$ can change over time $\Rightarrow$ *learning experts*

# Continuous Prediction (cont'd)

The definition of expert is so general that almost anything fits:

- $f_{i,t}$ can be a function of a *context* $x \Rightarrow$ *adaptive experts*
- $f_{i,t}$ can change over time $\Rightarrow$ *learning experts*
- $f_{i,t}$ is arbitrary $\Rightarrow$ experts can even form a *coalition against the learner*

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA
- The Continuous Prediction Game
- The Exponentially Weighted Average Forecaster
- Parameter Tuning
- Bounds for Small Losses

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

[Con]clusions

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \dots, f_{N,t}$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Predict ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)

$$\hat{p}_t = \frac{\sum_{i=1}^{N} w_{i,t-1} f_{i,t}}{W_{t-1}}$$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Predict ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)

$$\hat{p}_t = \frac{\sum_{i=1}^{N} w_{i,t-1} f_{i,t}}{W_{t-1}}$$

- Observe $y_t$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \dots, f_{N,t}$
- Predict ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)

$$\hat{p}_t = \frac{\sum_{i=1}^{N} w_{i,t-1} f_{i,t}}{W_{t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(\hat{p}_t, y_t)$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Predict $(W_{t-1} = \sum_{i=1}^N w_{i,t-1})$

$$\hat{p}_t = \frac{\sum_{i=1}^N w_{i,t-1} f_{i,t}}{W_{t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(\hat{p}_t, y_t)$
- Update

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \dots, f_{N,t}$
- Predict ($W_{t-1} = \sum_{i=1}^N w_{i,t-1}$)

$$\hat{p}_t = \frac{\sum_{i=1}^N w_{i,t-1} f_{i,t}}{W_{t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(\hat{p}_t, y_t)$
- Update (the weights are the exponential cumulative losses)

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

# The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Collect experts' predictions $f_{1,t}, \dots, f_{N,t}$
- Predict $(W_{t-1} = \sum_{i=1}^N w_{i,t-1})$

$$\hat{p}_t = \frac{\sum_{i=1}^N w_{i,t-1} f_{i,t}}{W_{t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(\hat{p}_t, y_t)$
- Update (the weights are the exponential cumulative losses)

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

Implement.: store and update the *normalized weights* $\hat{w}_{i,t} = w_{i,t}/W_t$.

# The Exponentially Weighted Average Forecaster (cont'd)

**Theorem**

*If $\mathcal{D}$ is a convex decision space and the loss function is bounded and convex in the first argument, then on any sequence $\mathbf{y}^n$, EWA($\eta$) satisfies*

$$R_n = L_n(\mathcal{A}; \mathbf{y}^n) - \min_i L_{i,n}(\mathbf{y}^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

# The Exponentially Weighted Average Forecaster (cont'd)

The proof is divided in three steps.

**Step 1: a lower bound on the log-ratio of cumulative weights**

$$\log \frac{W_{n+1}}{W_1} = \log W_{n+1} - \log W_1 = \log\Big(\sum_{i=1}^{N} w_{i,n+1}\Big) - \log N$$

# The Exponentially Weighted Average Forecaster (cont'd)

The proof is divided in three steps.

**Step 1: a lower bound on the log-ratio of cumulative weights**

$$\begin{aligned}\log \frac{W_{n+1}}{W_1} &= \log W_{n+1} - \log W_1 = \log\Big(\sum_{i=1}^{N} w_{i,n+1}\Big) - \log N \\ &\geq \log\Big(\max_{1\leq i\leq N} w_{i,n+1}\Big) - \log N\end{aligned}$$

# The Exponentially Weighted Average Forecaster (cont'd)

The proof is divided in three steps.

**Step 1: a lower bound on the log-ratio of cumulative weights**

$$\begin{aligned}
\log \frac{W_{n+1}}{W_1} &= \log W_{n+1} - \log W_1 = \log \Big( \sum_{i=1}^{N} w_{i,n+1} \Big) - \log N \\
&\geq \log \Big( \max_{1 \leq i \leq N} w_{i,n+1} \Big) - \log N \\
&= -\eta \min_{1 \leq i \leq N} \sum_{t=1}^{n} \ell(f_{i,t}, y_t) - \log N
\end{aligned}$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 2: an upper bound on the log-ratio of cumulative weights**

$$\log \frac{W_{t+1}}{W_t} = \log \Big( \sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \exp\big( -\eta\ell(f_{i,t}, y_t)\big)\Big)$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 2: an upper bound on the log-ratio of cumulative weights**

$$
\begin{aligned}
\log \frac{W_{t+1}}{W_t} &= \log \Big( \sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \exp\big( -\eta \ell(f_{i,t}, y_t) \big) \Big) \\
&= \log \big( \mathbb{E} \exp\big( -\eta \ell(f_{I_t,t}, y_t) \big) \big) \quad \text{(with } \mathbb{P}(I_t = i) = w_{i,t}/W_t \text{)}
\end{aligned}
$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 2: an upper bound on the log-ratio of cumulative weights**

$$
\begin{aligned}
\log \frac{W_{t+1}}{W_t} &= \log \Big( \sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \exp\big( -\eta \ell(f_{i,t}, y_t) \big) \Big) \\
&= \log \big( \mathbb{E} \exp\big( -\eta \ell(f_{I_t,t}, y_t) \big) \big) \quad \text{(with } \mathbb{P}(I_t = i) = w_{i,t}/W_t) \\
&\leq -\eta \mathbb{E} \ell(f_{I,t}, y_t) + \frac{\eta^2}{8} \quad \text{(Hoeffding's lemma)}
\end{aligned}
$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 2: an upper bound on the log-ratio of cumulative weights**

$$
\begin{aligned}
\log \frac{W_{t+1}}{W_t} &= \log\Big(\sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \exp\big(-\eta\ell(f_{i,t}, y_t)\big)\Big) \\
&= \log\big(\mathbb{E}\exp\big(-\eta\ell(f_{I_t,t}, y_t)\big)\big) \quad (\text{with } \mathbb{P}(I_t = i) = w_{i,t}/W_t) \\
&\leq -\eta\mathbb{E}\ell(f_{I,t}, y_t) + \frac{\eta^2}{8} \quad (\text{Hoeffding's lemma}) \\
&\leq -\eta\ell(\mathbb{E}f_{I,t}, y_t) + \frac{\eta^2}{8} \quad (\text{Jensen's inequality})
\end{aligned}
$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 2: an upper bound on the log-ratio of cumulative weights**

$$\begin{aligned}
\log \frac{W_{t+1}}{W_t} &= \log \Big( \sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \exp\big( -\eta \ell(f_{i,t}, y_t)\big)\Big) \\
&= \log \big( \mathbb{E} \exp \big( -\eta \ell(f_{I_t,t}, y_t)\big)\big) \quad (\text{with } \ \mathbb{P}(I_t = i) = w_{i,t}/W_t) \\
&\leq -\eta \mathbb{E} \ell(f_{I,t}, y_t) + \frac{\eta^2}{8} \quad (\text{Hoeffding's lemma}) \\
&\leq -\eta \ell(\mathbb{E} f_{I,t}, y_t) + \frac{\eta^2}{8} \quad (\text{Jensen's inequality}) \\
&= -\eta \ell(\hat{p}_t, y_t) + \frac{\eta^2}{8}
\end{aligned}$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 3: joint upper and lower bounds**
Notice that $\log \frac{W_{n+1}}{W_1} = \sum_{t=1}^{n} \log \frac{W_{t+1}}{W_t}$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 3: joint upper and lower bounds**
Notice that $\log \frac{W_{n+1}}{W_1} = \sum_{t=1}^{n} \log \frac{W_{t+1}}{W_t}$

$$\sum_{t=1}^{n} \log \frac{W_{t+1}}{W_t}$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 3: joint upper and lower bounds**
Notice that $\log \frac{W_{n+1}}{W_1} = \sum_{t=1}^n \log \frac{W_{t+1}}{W_t}$

$$\sum_{t=1}^n \log \frac{W_{t+1}}{W_t}$$

$$-\eta \min_{1 \leq i \leq N} \sum_{t=1}^n \ell(f_{i,t}, y_t) - \log N \leq \sum_{t=1}^n \log \frac{W_{t+1}}{W_t} \leq \sum_{t=1}^n \Big( -\eta \ell(\hat{p}_t, y_t) + \frac{\eta^2}{8} \Big)$$

# The Exponentially Weighted Average Forecaster (cont'd)

**Step 3: joint upper and lower bounds**
Notice that $\log \frac{W_{n+1}}{W_1} = \sum_{t=1}^n \log \frac{W_{t+1}}{W_t}$

$$\sum_{t=1}^n \log \frac{W_{t+1}}{W_t}$$

$$-\eta \min_{1\leq i\leq N} \sum_{t=1}^{n} \ell(f_{i,t}, y_t) - \log N \leq \sum_{t=1}^n \log \frac{W_{t+1}}{W_t} \leq \sum_{t=1}^{n} \Big( -\eta\ell(\hat{p}_t, y_t) + \frac{\eta^2}{8}\Big)$$

$$-\eta \min_{1\leq i\leq N} L_{i,n} - \log N \leq -\eta L_n(\mathcal{A}) + \frac{n\eta^2}{8}$$

The statement follows by reordering the terms. $\square$

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA
- The Continuous Prediction Game
- The Exponentially Weighted Average Forecaster
- Parameter Tuning
- Bounds for Small Losses

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Parameter Tuning

**Tuning**: how should we tune the parameter $\eta$?

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

# Parameter Tuning

**Tuning**: how should we tune the parameter $\eta$?

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

- Big $\eta$ = aggressive algorithm: *converge fast* to one expert but it could be *wrong*

# Parameter Tuning

**Tuning**: how should we tune the parameter $\eta$?

$$w_{i,t} = w_{i,t-1} \exp\big( - \eta \ell(f_{i,t}, y_t)\big)$$

- Big $\eta =$ aggressive algorithm: *converge fast* to one expert but it could be *wrong*
- Small $\eta =$ conservative algorithm: *does not converge to the wrong expert* but it could take a *long time*

# Parameter Tuning (cont'd)

**Tuning**: how should we tune the parameter $\eta$?

$$w_{i,t} = w_{i,t-1} \exp\big( -\eta \ell(f_{i,t}, y_t)\big)$$

# Parameter Tuning (cont'd)

**Tuning**: how should we tune the parameter $\eta$?

$$w_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

$$R_n(EWA) \leq \underbrace{\frac{\log N}{\eta}}_{\text{big!}} + \underbrace{\frac{\eta n}{8}}_{\text{small!}}$$

# Parameter Tuning (cont'd)

**Tuning**: If we know the horizon $n$, then by setting $\eta = \sqrt{\frac{8 \log N}{n}}$

# Parameter Tuning (cont'd)

**Tuning**: If we know the horizon $n$, then by setting $\eta = \sqrt{\frac{8 \log N}{n}}$

$$R_n(EWA) \leq \sqrt{\frac{n}{2} \log N}$$

# Parameter Tuning (cont'd)

**Tuning**: If we know the horizon $n$, then by setting $\eta = \sqrt{\frac{8 \log N}{n}}$

$$R_n(EWA) \leq \sqrt{\frac{n}{2} \log N}$$

- Logarithmic dependency on $N$
  $\Rightarrow$ add many experts, no problem!

# Parameter Tuning (cont'd)

**Tuning**: If we know the horizon $n$, then by setting $\eta = \sqrt{\frac{8 \log N}{n}}$

$$R_n(EWA) \leq \sqrt{\frac{n}{2} \log N}$$

- Logarithmic dependency on $N$
  $\Rightarrow$ add many experts, no problem!
- Per–step regret $R_n/n = \sqrt{1/n} \rightarrow 0$

# Parameter Tuning (cont'd)

**Tuning**: If we know the horizon $n$, then by setting $\eta = \sqrt{\frac{8 \log N}{n}}$

$$R_n(EWA) \leq \sqrt{\frac{n}{2} \log N}$$

- Logarithmic dependency on $N$
  $\Rightarrow$ add many experts, no problem!
- Per–step regret $R_n/n = \sqrt{1/n} \rightarrow 0$
  $\Rightarrow$ EWA is asymptotically as good as the best expert!

# Parameter Tuning (cont'd)

**Problem**: Sometimes $n$ is unknown (or it does not exist at all)

# Parameter Tuning (cont'd)

**Problem**: Sometimes $n$ is unknown (or it does not exist at all)

**Solution**: set $\eta_t = 2\sqrt{\frac{\log N}{t}}$ and

$$R_n(EWA) \leq \sqrt{n \log N}$$

# A Comparison with SLT results

Bound for ***batch*** binary classification with *$N$ hypotheses* on data *i.i.d. from $\mathcal{P}$*

$$R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) \leq O\Big(\sqrt{\frac{\log N/\delta}{n}}\Big)$$

if the observations are i.i.d. from a stationary distribution $\mathcal{P}$

# A Comparison with SLT results

Bound for ***batch*** binary classification with *$N$ hypotheses* on data *i.i.d. from $\mathcal{P}$*

$$n\big(R(\hat{h};\mathcal{P}) - \min_{h\in\mathcal{H}} R(h;\mathcal{P})\big) \leq O\Big(\sqrt{n\log N/\delta}\Big)$$

if the observations are i.i.d. from a stationary distribution $\mathcal{P}$

# A Comparison with SLT results

Bound for ***batch*** binary classification with *N hypotheses* on data *i.i.d. from* $\mathcal{P}$

$$n\Big(\mathbb{E}_{x,y}[\ell(\hat{h}(x),y)] - \min_{h\in\mathcal{H}} \mathbb{E}_{x,y}[\ell(h(x),y)]\Big) \le O\Big(\sqrt{n \log N/\delta}\Big)$$

if the observations are i.i.d. from a stationary distribution $\mathcal{P}$

# A Comparison with SLT results

Bound for ***batch*** binary classification with *N hypotheses* on data *i.i.d. from* $\mathcal{P}$

$$\mathbb{E}_{x,y}[n\ell(\hat{h}(x),y)] - \min_{h\in\mathcal{H}} \mathbb{E}_{x,y}[n\ell(h(x),y)]\Big) \leq O\Big(\sqrt{n \log N/\delta}\Big)$$

if the observations are i.i.d. from a stationary distribution $\mathcal{P}$

# A Comparison with SLT results

Bound for ***batch*** binary classification with *$N$ hypotheses* on data *i.i.d. from $\mathcal{P}$*

$$\mathbb{E}_{x,y}[n\ell(\hat{h}(x), y)] - \min_{h \in \mathcal{H}} \mathbb{E}_{x,y}[n\ell(h(x), y)]\big) \leq O\Big(\sqrt{n \log N/\delta}\Big)$$

if the observations are i.i.d. from a stationary distribution $\mathcal{P}$

Bound for ***online*** binary classification with *$N$ experts* on *any sequence $\mathbf{y}^n$*

$$\sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_i \sum_{t=1}^{n} \ell(f_{i,t}, y_t) \leq \sqrt{n \log N}$$

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA
- The Continuous Prediction Game
- The Exponentially Weighted Average Forecaster
- Parameter Tuning
- Bounds for Small Losses

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

lusions

# An Alternative Bound (for Small Losses)

**Question**: What if the best expert is ***really*** good? (i.e., $L_n^* = \min_i L_{i,n}$ is small)

# An Alternative Bound (for Small Losses) (cont'd)

**Theorem**

*If $\mathcal{D}$ is a convex decision space and the loss function is bounded and convex in the first argument. Let $L_n^* = \min_i L_{i,n}$, then on any sequence $\mathbf{y}^n$, EWA($\eta$) satisfies*

$$L_n(\mathcal{A}) \leq \frac{\eta L_n^* + \log N}{1 - \exp^{-\eta}}$$

# An Alternative Bound (for Small Losses) (cont'd)

## Corollary

*If $\eta = 1$ (aggressive algorithm)*

$$L_n(\mathcal{A}) \leq \frac{e}{e-1}\left(L_n^* + \log N\right) = L_n^* + \frac{1}{e-1}L_n^* + \frac{e}{e-1}\log N$$

# An Alternative Bound (for Small Losses) (cont'd)

**Corollary**

*If $\eta = 1$ (aggressive algorithm)*

$$L_n(\mathcal{A}) \leq \frac{e}{e-1}\big(L_n^* + \log N\big) = L_n^* + \frac{1}{e-1}L_n^* + \frac{e}{e-1}\log N$$

- If $L_n^*$ is small (i.e., $L_n^* \ll \sqrt{n}$) it is much *better* than the previous bound

# An Alternative Bound (for Small Losses) (cont'd)

## Corollary

*If $\eta = 1$ (aggressive algorithm)*

$$L_n(\mathcal{A}) \leq \frac{e}{e-1}\big(L_n^* + \log N\big) = L_n^* + \frac{1}{e-1}L_n^* + \frac{e}{e-1}\log N$$

- If $L_n^*$ is small (i.e., $L_n^* \ll \sqrt{n}$) it is much *better* than the previous bound
- If $L_n^*$ is not small (i.e., $L_n^* > \sqrt{n}$) it is much *worse* than the previous bound

# An Alternative Bound (for Small Losses) (cont'd)

**Corollary**

*If $\eta = 1$ (aggressive algorithm)*

$$L_n(\mathcal{A}) \leq \frac{e}{e-1}\big(L_n^* + \log N\big) = L_n^* + \frac{1}{e-1}L_n^* + \frac{e}{e-1}\log N$$

- If $L_n^*$ is small (i.e., $L_n^* \ll \sqrt{n}$) it is much *better* than the previous bound
- If $L_n^*$ is not small (i.e., $L_n^* > \sqrt{n}$) it is much *worse* than the previous bound
- If $L_n^* = 0$ we have (almost) the same performance as the *Halving algorithm*

# An Alternative Bound (for Small Losses) (cont'd)

## Corollary

*If we optimally tune* $\eta = \log(1 + \sqrt{(2 \log N)/L_n^*})$

$$L_n(\mathcal{A}) \leq L_n^* + \sqrt{2L_n^* \log N} + \log N$$

# An Alternative Bound (for Small Losses) (cont'd)

## Corollary

*If we optimally tune* $\eta = \log(1 + \sqrt{(2\log N)/L_n^*})$

$$L_n(\mathcal{A}) \leq L_n^* + \sqrt{2L_n^* \log N} + \log N$$

**Problem**: the performance of the best expert is usually not known...

Algorithm adapting to the **complexity** of the problem?

# An Alternative Bound (for Small Losses) (cont'd)

**Corollary**

*If we optimally tune* $\eta = \log(1 + \sqrt{(2 \log N)/L_n^*})$

$$L_n(\mathcal{A}) \leq L_n^* + \sqrt{2L_n^* \log N} + \log N$$

**Problem**: the performance of the best expert is usually not known...

Algorithm adapting to the **complexity** of the problem?

*Almost... (see NIPS this year)*

# Outline


Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA
- The Discrete Prediction Game
- A Note on Lower Bounds

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA
- The Discrete Prediction Game
- A Note on Lower Bounds

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Discrete Prediction

- Outcome space $\mathcal{Y}$ is discrete (with $|Y| \geq 2$)
- Decision space $\mathcal{D} = \mathcal{Y}$
- Loss function $\ell(p, y) = \mathbb{I}\{p \neq y\}$

# Discrete Prediction (cont'd)

- Experts $f_{1,t}, \ldots, f_{N,t}$
- The performance measure: the (expert) regret

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_{1 \leq i \leq N} \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

# Discrete Prediction (cont'd)

**Remark**: everything is almost the same as in the continuous prediction, so it should be easy!

# Discrete Prediction (cont'd)

**Remark**: everything is almost the same as in the continuous prediction, so it should be easy! ***No***

# Discrete Prediction (cont'd)

**Example**: Two experts: $f_{1,t} = 0$ and $f_{2,t} = 1$ at any $t$, then

# Discrete Prediction (cont'd)

**Example**: Two experts: $f_{1,t} = 0$ and $f_{2,t} = 1$ at any $t$, then

- For any sequence $\mathbf{y}^n = (y_1, \ldots, y_n) \in \{0, 1\}^n$, there exists an experts $i$ such that

$$L_{i,n} = \sum_{t=1}^{n} \ell(f_{i,t}, y_t) \geq n/2$$

# Discrete Prediction (cont'd)

**Example**: Two experts: $f_{1,t} = 0$ and $f_{2,t} = 1$ at any $t$, then

- For any sequence $\mathbf{y}^n = (y_1, \ldots, y_n) \in \{0, 1\}^n$, there exists an experts $i$ such that

$$L_{i,n} = \sum_{t=1}^{n} \ell(f_{i,t}, y_t) \geq n/2$$

- For any algorithm $\mathcal{A}$, there exists a sequence $\mathbf{y}^n(\mathcal{A})$ such that

$$L_n(\mathcal{A}) = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t(\mathcal{A})) = n$$

# Discrete Prediction (cont'd)

Let's (adversarially) construct the sequence $\mathbf{y}^n(\mathcal{A})$.

- At time 1, the adversary sets $y_1(\mathcal{A}) = 1 - \hat{p}_1$ (for a fixed algorithm $\mathcal{A}$ this is always possible)

# Discrete Prediction (cont'd)

Let's (adversarially) construct the sequence $\mathbf{y}^n(\mathcal{A})$.

- At time 1, the adversary sets $y_1(\mathcal{A}) = 1 - \hat{p}_1$ (for a fixed algorithm $\mathcal{A}$ this is always possible)
- At time $t$, the algorithm chooses $\hat{p}_t$ on the basis of $(y_1(\mathcal{A}), \ldots, y_{t-1}(\mathcal{A}))$ (in a predictable way)

# Discrete Prediction (cont'd)

Let's (adversarially) construct the sequence $\mathbf{y}^n(\mathcal{A})$.

- At time 1, the adversary sets $y_1(\mathcal{A}) = 1 - \hat{p}_1$ (for a fixed algorithm $\mathcal{A}$ this is always possible)
- At time $t$, the algorithm chooses $\hat{p}_t$ on the basis of $(y_1(\mathcal{A}), \ldots, y_{t-1}(\mathcal{A}))$ (in a predictable way)
- At time $t$, the adversary sets $y_t(\mathcal{A}) = 1 - \hat{p}_t$

# Discrete Prediction (cont'd)

**Theorem**

*In the discrete prediction problem, for any deterministic algorithm $\mathcal{A}$, the worst case regret is*

$$R_n(\mathcal{A}) \geq \frac{n}{2}$$

# Discrete Prediction (cont'd)

**Theorem**

*In the discrete prediction problem, for any deterministic algorithm $\mathcal{A}$, the worst case regret is*

$$R_n(\mathcal{A}) \geq \frac{n}{2}$$

# Discrete Prediction (cont'd)

**Theorem**

*In the discrete prediction problem, for any deterministic algorithm $\mathcal{A}$, the worst case regret is*

$$R_n(\mathcal{A}) \geq \frac{n}{2}$$

**Solution**: *let's randomize!*

# Discrete Prediction (cont'd)

**Problem**: how do we *randomize* over experts without *loosing in performance*?

# Discrete Prediction (cont'd)

**Problem**: how do we *randomize* over experts without *loosing in performance*?
**Solution**: use the Exponentially Weighted Average forecaster!

# Discrete Prediction (cont'd)

We first construct a *fictitious* continuous prediction problem where we can apply the EWA:

- $\mathcal{D}' = \{p \in [0,1]^N \ : \sum_{i=1}^N p_i = 1\} \Rightarrow$ convex

# Discrete Prediction (cont'd)

We first construct a *fictitious* continuous prediction problem where we can apply the EWA:

- $\mathcal{D}' = \{p \in [0,1]^N \ : \sum_{i=1}^N p_i = 1\} \Rightarrow$ convex
- $Y' = Y \times \mathcal{D}^N$

# Discrete Prediction (cont'd)

We first construct a *fictitious* continuous prediction problem where we can apply the EWA:

- $\mathcal{D}' = \{p \in [0,1]^N \ : \sum_{i=1}^N p_i = 1\} \Rightarrow$ convex
- $Y' = Y \times \mathcal{D}^N$
- $\ell'(p, (y, f_1, \dots, f_N)) = \sum_{i=1}^N p_i \ell(f_i, y) \Rightarrow$ convex and bounded

# Discrete Prediction (cont'd)

We first construct a *fictitious* continuous prediction problem where we can apply the EWA:

- $\mathcal{D}' = \{p \in [0,1]^N \ : \sum_{i=1}^N p_i = 1\} \Rightarrow$ convex
- $Y' = Y \times \mathcal{D}^N$
- $\ell'(p, (y, f_1, \ldots, f_N)) = \sum_{i=1}^N p_i \ell(f_i, y) \Rightarrow$ convex and bounded
- $f'_{i,t} = e_i$, with $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ with $i$-th coordinate equal to 1

# Discrete Prediction (cont'd)

We first construct a *fictitious* continuous prediction problem where we can apply the EWA:

- $\mathcal{D}' = \{p \in [0,1]^N \ : \sum_{i=1}^N p_i = 1\} \Rightarrow$ convex
- $Y' = Y \times \mathcal{D}^N$
- $\ell'(p, (y, f_1, \dots, f_N)) = \sum_{i=1}^N p_i \ell(f_i, y) \Rightarrow$ convex and bounded
- $f'_{i,t} = e_i$, with $e_i = (0, \dots, 0, 1, 0, \dots, 0)^\top$ with $i$-th coordinate equal to 1
- $y'_t = (y_t, f_{1,t}, \dots, f_{N,t})$

# Discrete Prediction (cont'd)

We notice that

$$\ell'(f'_{i,t}, y'_t) = \ell'(e_i, (y_t, f_{1,t}, \dots, f_{N,t})) = \ell(f_{i,t}, y_t)$$

Thus

$$L_{i,t} = \sum_{s=1}^{t} \ell(f_{i,s}, y_s) = \sum_{s=1}^{t} \ell'(f'_{i,s}, y'_s)$$

# Discrete Prediction (cont'd)

At each round $t$ of the *fictitious continuos* problem the algorithm returns

$$\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$$

# Discrete Prediction (cont'd)

At each round $t$ of the *fictitious continuos* problem the algorithm returns

$$\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$$

At each round $t$ of the *real discrete* problem the algorithm returns (***at random***)

$$I_t \sim \hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$$

# Discrete Prediction (cont'd)

At each round $t$ of the *fictitious continuos* problem the algorithm returns

$$\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$$

At each round $t$ of the *real discrete* problem the algorithm returns (***at random***)

$$I_t \sim \hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$$

and in expectation

$$\mathbb{E}[\ell(f_{I_t}, y_t)] = \sum_{t=1}^{N} \hat{p}_{i,t} \ell(f_{i,t}, y_t) = \ell'(\hat{p}_t, (y_t, f_{1,t}, \ldots, f_{N,t})) = \ell'(\hat{p}_t, y'_t)$$

# Discrete Prediction (cont'd)

The performance is

$$L'_n(\mathcal{A}) = \sum_{t=1}^{n} \ell'(\hat{p}_t, y'_t) = \mathbb{E}\Big[\sum_{t=1}^{n} \ell(f_{I_t,t}, y_t)\Big] = \mathbb{E}[L_n(\mathcal{A})]$$

# Discrete Prediction (cont'd)

| Discrete | Continuous | |
|---|---|---|
| $\ell(f_i, y)$ | $\ell'(p, y') = \sum_{i=1}^{N} p_i \ell(f_i, y)$ | |
| $\ell(f_{i,t}, y_t)$ | $\ell'(f'_{i,t}, y'_t)$ | |
| $\mathbb{E}[\ell(f_{I_t}, y_t)]$ | $\ell'(\hat{p}_t, y'_t)$ | |
| $\mathbb{E}[L_n(\mathcal{A})]$ | $L'_n(\mathcal{A})$ | |

# Discrete Prediction (cont'd)

| Discrete | Continuous | |
|---|---|---|
| $\ell(f_i, y)$ | $\ell'(p, y') = \sum_{i=1}^N p_i \ell(f_i, y)$ | |
| $\ell(f_{i,t}, y_t)$ | $\ell'(f'_{i,t}, y'_t)$ | cumulative losses coincide |
| $\mathbb{E}[\ell(f_{I_t}, y_t)]$ | $\ell'(\hat{p}_t, y'_t)$ | |
| $\mathbb{E}[L_n(\mathcal{A})]$ | $L'_n(\mathcal{A})$ | |

# Discrete Prediction (cont'd)

| Discrete | Continuous | |
|---|---|---|
| $\ell(f_i, y)$ | $\ell'(p, y') = \sum_{i=1}^{N} p_i \ell(f_i, y)$ | |
| $\ell(f_{i,t}, y_t)$ | $\ell'(f'_{i,t}, y'_t)$ | cumulative losses coincide |
| $\mathbb{E}[\ell(f_{I_t}, y_t)]$ | $\ell'(\hat{p}_t, y'_t)$ | coincide in expectation |
| $\mathbb{E}[L_n(\mathcal{A})]$ | $L'_n(\mathcal{A})$ | |

# Discrete Prediction (cont'd)

| Discrete | Continuous | |
|---|---|---|
| $\ell(f_i, y)$ | $\ell'(p, y') = \sum_{i=1}^{N} p_i \ell(f_i, y)$ | |
| $\ell(f_{i,t}, y_t)$ | $\ell'(f'_{i,t}, y'_t)$ | cumulative losses coincide |
| $\mathbb{E}[\ell(f_{I_t}, y_t)]$ | $\ell'(\hat{p}_t, y'_t)$ | coincide in expectation |
| $\mathbb{E}[L_n(\mathcal{A})]$ | $L'_n(\mathcal{A})$ | coincide in expectation |

# Discrete Prediction (cont'd)

| Discrete | Continuous | |
|---|---|---|
| $\ell(f_i, y)$ | $\ell'(p, y') = \sum_{i=1}^{N} p_i \ell(f_i, y)$ | |
| $\ell(f_{i,t}, y_t)$ | $\ell'(f'_{i,t}, y'_t)$ | cumulative losses coincide |
| $\mathbb{E}[\ell(f_{I_t}, y_t)]$ | $\ell'(\hat{p}_t, y'_t)$ | coincide in expectation |
| $\mathbb{E}[L_n(\mathcal{A})]$ | $L'_n(\mathcal{A})$ | coincide in expectation |

# Discrete Prediction (cont'd)

**Theorem**

*If $\mathcal{D}'$ is a convex decision space and the loss function $\ell'$ is bounded and convex in the first argument, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

$$R'_n = L'_n(\mathcal{A}; \mathbf{y}'^n) - \min_i L'_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

# Discrete Prediction (cont'd)

## Theorem

*If $\mathcal{D}'$ is a convex decision space and the loss function $\ell'$ is bounded and convex in the first argument, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

$$R'_n = L'_n(\mathcal{A}; \mathbf{y}'^n) - \min_i L_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

# Discrete Prediction (cont'd)

## Theorem

*If $\mathcal{D}'$ is a convex decision space and the loss function $\ell'$ is bounded and convex in the first argument, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

$$R'_n = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

# Discrete Prediction (cont'd)

**Theorem**

*If $\mathcal{D}'$ is a convex decision space and the loss function $\ell'$ is bounded and convex in the first argument, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

$$\mathbb{E}[R_n] = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

# Discrete Prediction (cont'd)

**Theorem**

*If $\mathcal{D}'$ is a convex decision space and the loss function $\ell'$ is bounded and convex in the first argument, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

# Discrete Prediction (cont'd)

**Theorem**

*If $\mathcal{D}$ is a is a discrete space and $\ell$ is any loss function, then on any sequence $\mathbf{y}'^n$, EWA($\eta$) satisfies*

# Discrete Prediction (cont'd)

## Theorem

*If $\mathcal{D} = \mathcal{Y}$ are discrete spaces and $\ell$ is any loss function, then on any sequence $\mathbf{y}'^n$, the randomized EWA($\eta$) satisfies*

$$\mathbb{E}[R_n] = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

*and*

$$\mathbb{E}[R_n] = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \sqrt{\frac{n}{2} \log N}.$$

*if $\eta$ is properly tuned.*

# Discrete Prediction (cont'd)

## Theorem

*If $\mathcal{D} = \mathcal{Y}$ are discrete spaces and $\ell$ is any loss function, then on any sequence $\mathbf{y}'^n$, the randomized EWA($\eta$) satisfies*

$$\mathbb{E}[R_n] = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \frac{\log N}{\eta} + \frac{\eta n}{8}.$$

*and*

$$\mathbb{E}[R_n] = \mathbb{E}[L_n(\mathcal{A}; \mathbf{y}'^n)] - \min_i L_{i,n}(\mathbf{y}'^n) \leq \sqrt{\frac{n}{2} \log N}.$$

*if $\eta$ is properly tuned.*

**Problem**: interesting but this holds only on average, does it mean that from time to time the algorithm can perform arbitrarily bad?

# Discrete Prediction (cont'd)

**Solution**: do you remember the Chernoff-Hoeffding bound?

$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t - \sum_{t=1}^{n} \mathbb{E}[X_t] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

# Discrete Prediction (cont'd)

**Solution**: do you remember the Chernoff-Hoeffding bound?

$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t - \sum_{t=1}^{n} \mathbb{E}[X_t] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

$\Rightarrow$

$$\mathbb{P}\Big[\sum_{t=1}^{n} \ell(f_{I_t,t}, y_t) - \sum_{t=1}^{n} \mathbb{E}[\ell(f_{I_t,t}, y_t)] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

# Discrete Prediction (cont'd)

**Solution**: do you remember the Chernoff-Hoeffding bound?

$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t - \sum_{t=1}^{n} \mathbb{E}[X_t] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

$\Rightarrow$

$$\mathbb{P}\Big[\sum_{t=1}^{n} \ell(f_{I_t,t}, y_t) - \sum_{t=1}^{n} \mathbb{E}[\ell(f_{I_t,t}, y_t)] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

$\Rightarrow$

$$\mathbb{P}\Big[L_n(\mathcal{A}) - \mathbb{E}[L_n(\mathcal{A})] > \varepsilon\Big] \leq \exp\big(-2\varepsilon^2/n\big)$$

# Discrete Prediction (cont'd)

**Theorem**

*If $\mathcal{D} = \mathcal{Y}$ are discrete spaces and $\ell$ is any loss function, then on any sequence $\mathbf{y}'^n$, the randomized EWA($\eta$) satisfies*

$$R_n = L_n(\mathcal{A}; \mathbf{y}^n) - \min_i L_{i,n}(\mathbf{y}^n) \leq \sqrt{\frac{n}{2} \log N} + \sqrt{\frac{n}{2} \log \frac{1}{\delta}}$$

*with probability $1 - \delta$.*

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA
- The Discrete Prediction Game
- A Note on Lower Bounds

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Lower Bounds

**Question**: EWA($\eta$) seems good but **I** am sure that **my** algorithm can *do better*!

# Lower Bounds

**Question**: EWA($\eta$) seems good but I am sure that **my** algorithm can *do better*!

**Answer**: don't even try... EWA is the *best possible algorithm*!
Informally:

$$\inf_{\mathcal{A}} \sup_{\mathbf{y}^n} R_n(\mathcal{A}; \mathbf{y}^n) \geq \sqrt{\frac{n}{2} \log N}$$

# Lower Bounds

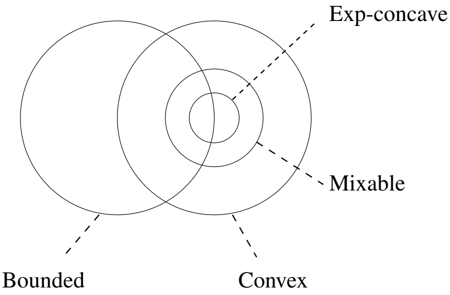
**Question**: EWA($\eta$) seems good but I am sure that **my** algorithm can *do better*!

**Answer**: don't even try... EWA is the *best possible algorithm*! Informally:

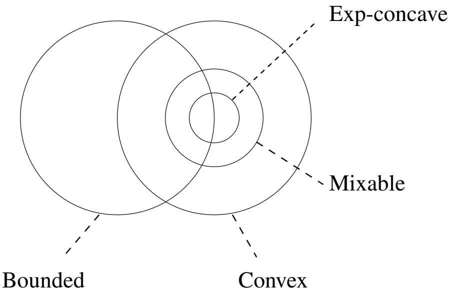
$$\inf_{\mathcal{A}} \sup_{\mathbf{y}^n} R_n(\mathcal{A}; \mathbf{y}^n) \geq \sqrt{\frac{n}{2} \log N}$$

for some losses...

# Lower Bounds (cont'd)

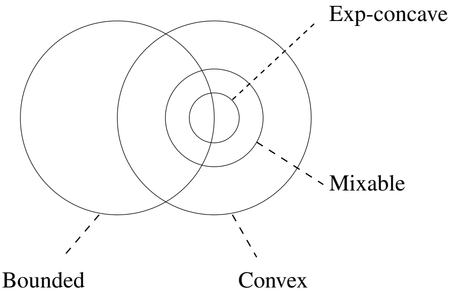


# Lower Bounds (cont'd)



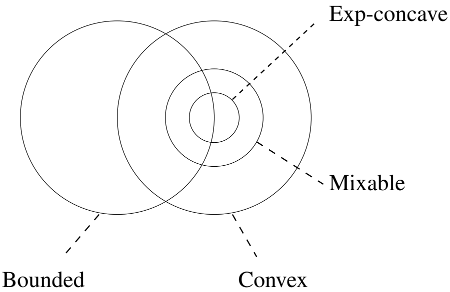

- Bounded and convex: EWA is optimal with regret $O(\sqrt{n \log N})$

# Lower Bounds (cont'd)



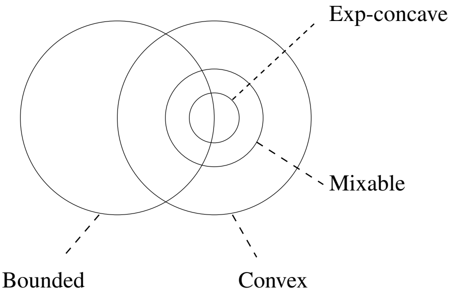

- Bounded and convex: EWA is optimal with regret $O(\sqrt{n \log N})$
- Mixable: optimal regret $c \log N$ but not (always) achieved EWA

# Lower Bounds (cont'd)




- Bounded and convex: EWA is optimal with regret $O(\sqrt{n \log N})$
- Mixable: optimal regret $c \log N$ but not (always) achieved EWA
- Exp-concave: EWA is optimal with regret $c \log N$

# Lower Bounds (cont'd)




- Bounded and convex: EWA is optimal with regret $O(\sqrt{n \log N})$
- Mixable: optimal regret $c \log N$ but not (always) achieved EWA
- Exp-concave: EWA is optimal with regret $c \log N$
- Non-convex: EWA is optimal in discrete prediction

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts
- Tracking the Best Expert
- Tree Experts
- Shortest Path Problem
- Infinite Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts
- Tracking the Best Expert
- Tree Experts
- Shortest Path Problem
- Infinite Experts

$$ How to Make Money with Online Learning $$

Conclusions

# A Remark on the Regret

$$R_n = L_n(\mathcal{A}) - \min_i L_{i,n}$$

# A Remark on the Regret

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_i \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

# A Remark on the Regret

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_i \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

**Remark**: algorithm competes against the best *fixed* expert

# A Remark on the Regret

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_i \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

**Remark**: algorithm competes against the best *fixed* expert
**Problem**: what if the *good* expert *changes over time*?

# A Remark on the Regret (cont'd)

**Question**: why do not design an algorithm to compete against the best *changing* expert?

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \min_{i} \sum_{t=1}^{n} \ell(f_{i,t}, y_t)$$

# A Remark on the Regret (cont'd)

**Question**: why do not design an algorithm to compete against the best *changing* expert?

$$R_n = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \sum_{t=1}^{n} \min_i \ell(f_{i,t}, y_t)$$

# Switching Experts

A *switching* compound expert $\sigma$ is

$$\sigma \in \{1, \ldots, N\}^n$$

# Switching Experts

A *switching* compound expert $\sigma$ is

$$\sigma \in \{1, \ldots, N\}^n$$

At each round $t$ it chooses expert $\sigma_t$ and *cumulate a loss*

$$L_{\sigma,n} = \sum_{t=1}^{n} \ell(f_{\sigma_t,t}, y_t)$$

# Switching Experts

A *switching* compound expert $\sigma$ is

$$\sigma \in \{1, \ldots, N\}^n$$

At each round $t$ it chooses expert $\sigma_t$ and *cumulate a loss*

$$L_{\sigma,n} = \sum_{t=1}^{n} \ell(f_{\sigma_t,t}, y_t)$$

*Class of switching experts* $B \subseteq \{1, \ldots, N\}^n$
We refer to the others as *base experts*.

# Switching Experts (cont'd)

**Problem**: At each round $t$ the learner takes the action suggested by the switching expert $\hat{\sigma}_t$, thus cumulating

$$L_n(\mathcal{A}) = \sum_{t=1}^{n} \ell(f_{\hat{\sigma}_t,t}, y_t)$$

# Switching Experts (cont'd)

**Problem**: At each round $t$ the learner takes the action suggested by the switching expert $\hat{\sigma}_t$, thus cumulating

$$L_n(\mathcal{A}) = \sum_{t=1}^{n} \ell(f_{\hat{\sigma}_t,t}, y_t)$$

The regret of $\mathcal{A}$ w.r.t. switching experts in $B$ is

$$R_n = L_n(\mathcal{A}) - \min_i L_{i,n}$$

# Switching Experts (cont'd)

**Problem**: At each round $t$ the learner takes the action suggested by the switching expert $\hat{\sigma}_t$, thus cumulating

$$L_n(\mathcal{A}) = \sum_{t=1}^{n} \ell(f_{\hat{\sigma}_t,t}, y_t)$$

The regret of $\mathcal{A}$ w.r.t. switching experts in $B$ is

$$R_n = L_n(\mathcal{A}) - \min_{\sigma \in B} L_{\sigma,n}$$

# Switching Experts (cont'd)

**Problem**: At each round $t$ the learner takes the action suggested by the switching expert $\hat{\sigma}_t$, thus cumulating

$$L_n(\mathcal{A}) = \sum_{t=1}^{n} \ell(f_{\hat{\sigma}_t,t}, y_t)$$

The regret of $\mathcal{A}$ w.r.t. switching experts in $B$ is

$$R_n = L_n(\mathcal{A}) - \min_{\sigma \in B} L_{\sigma,n}$$

**Solution**: use the EWA on the set of *meta*-experts $B$!

# Switching Experts (cont'd)

**Corollary**

*In online discrete prediction, the EWA($\eta$) run on the class $B$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n = L_n(\mathcal{A}) - \min_{\sigma \in B} L_{\sigma,n} \leq \sqrt{\frac{n}{2} \log |B|}$$

# Switching Experts (cont'd)

## Corollary

*In online discrete prediction, the EWA($\eta$) run on the class $B$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n = L_n(\mathcal{A}) - \min_{\sigma \in B} L_{\sigma,n} \leq \sqrt{\frac{n}{2} \log |B|}$$

**Problem**: if $B = \{1, \ldots, N\}^n$ then $|B| = N^n$ and

$$R_n \leq \sqrt{\frac{n}{2} \log |B|} = O(n)$$

$\Rightarrow$ sad facts of life... we cannot compete against the sequence of best experts

# Switching Experts (cont'd)

**Question**: what if we limit the *number of switches* of the switching experts to $m$?

$$s(\sigma) = \sum_{t=1}^{n} \mathbb{I}\left\{\sigma_{t-1} \neq \sigma_t\right\}$$

# Switching Experts (cont'd)

**Question**: what if we limit the *number of switches* of the switching experts to $m$?

$$s(\sigma) = \sum_{t=1}^{n} \mathbb{I}\{\sigma_{t-1} \neq \sigma_t\}$$

$$B_{n,m} = \{\sigma | \, s(\sigma) \leq m\}$$

# Switching Experts (cont'd)

**Question**: what if we limit the *number of switches* of the switching experts to $m$?

$$s(\sigma) = \sum_{t=1}^{n} \mathbb{I}\left\{\sigma_{t-1} \neq \sigma_t\right\}$$

$$B_{n,m} = \{\sigma |\; s(\sigma) \leq m\}$$

$$|B_{n,m}| = \sum_{k=0}^{m} \binom{n-1}{k} N(N-1)^k \leq N^{m+1} \exp\Big((n-1)H\big(\frac{m}{n-1}\big)\Big)$$

with $H(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function.

# Switching Experts (cont'd)

## Corollary

*In online discrete prediction, the EWA($\eta$) run on the class $B_{n,m}$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n \leq \sqrt{\frac{n}{2}\Big((m+1)\log N + (n-1)H\big(\frac{m}{n-1}\big)\Big)}$$

# Switching Experts (cont'd)

**Corollary**

*In online discrete prediction, the EWA($\eta$) run on the class $B_{n,m}$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n \leq \sqrt{\frac{n}{2}\Big((m+1)\log N + (n-1)H\big(\frac{m}{n-1}\big)\Big)}$$

# Switching Experts (cont'd)

**Corollary**

*In online discrete prediction, the EWA($\eta$) run on the class $B_{n,m}$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n \leq \sqrt{\frac{n}{2}\Big((m+1)\log N + (n-1)H\big(\frac{m}{n-1}\big)\Big)}$$

**Problem**: not bad, but the EWA should maintain and update $|B_{n,m}|$ weights... *unfeasible*!

# Switching Experts (cont'd)

## Corollary

*In online discrete prediction, the EWA($\eta$) run on the class $B_{n,m}$ of switching experts achieves (with a suitable choice of $\eta$)*

$$R_n \leq \sqrt{\frac{n}{2}\Big((m+1)\log N + (n-1)H(\frac{m}{n-1})\Big)}$$

**Problem**: not bad, but the EWA should maintain and update $|B_{n,m}|$ weights... *unfeasible*!
**Objective**: an *efficient* EWA algorithm which maintains as many weights as the N *base* experts

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Randomize according to

$$I_t \sim \hat{p}_{i,t} = \frac{w_{i,t-1} f_{i,t}}{\sum_{j=1}^{N} w_{j,t-1}}$$

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Randomize according to

$$I_t \sim \hat{p}_{i,t} = \frac{w_{i,t-1} f_{i,t}}{\sum_{j=1}^{N} w_{j,t-1}}$$

- Observe $y_t$

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Randomize according to

$$I_t \sim \hat{p}_{i,t} = \frac{w_{i,t-1} f_{i,t}}{\sum_{j=1}^{N} w_{j,t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(f_{I_t,t}, y_t)$

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \dots, f_{N,t}$
- Randomize according to

$$I_t \sim \hat{p}_{i,t} = \frac{w_{i,t-1} f_{i,t}}{\sum_{j=1}^{N} w_{j,t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(f_{I_t,t}, y_t)$
- Compute

$$v_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

# The Fixed-Share Forecaster

Initialize the weights $w_{i,0} = 1/N$

- Collect experts' predictions $f_{1,t}, \ldots, f_{N,t}$
- Randomize according to

$$I_t \sim \hat{p}_{i,t} = \frac{w_{i,t-1} f_{i,t}}{\sum_{j=1}^N w_{j,t-1}}$$

- Observe $y_t$
- Suffer a loss $\ell(f_{I_t,t}, y_t)$
- Compute

$$v_{i,t} = w_{i,t-1} \exp\big(-\eta \ell(f_{i,t}, y_t)\big)$$

- Update (with $W_t = \sum_i v_{i,t}$)

$$w_{i,t} = \alpha \frac{W_t}{N} + (1-\alpha) v_{i,t}$$

# The Fixed-Share Forecaster (cont'd)

**Intuition**: $\alpha$ encodes a *belief* on the switching frequency

$$w_{i,t} = \alpha \frac{W_t}{N} + (1-\alpha) v_{i,t}$$

# The Fixed-Share Forecaster (cont'd)

**Details**: everything starts from a non–uniform belief over the class $B$ of *all* the possible switching strategies $\sigma = (\sigma_1, \dots, \sigma_n)$

$$w'_0(\sigma) = \frac{1}{N}\left(\frac{\alpha}{N}\right)^{s(\sigma)}\left(1 - \alpha + \frac{\alpha}{N}\right)^{n-s(\sigma)}$$

# The Fixed-Share Forecaster (cont'd)

**Details**: everything starts from a non–uniform belief over the class $B$ of *all* the possible switching strategies $\sigma = (\sigma_1, \ldots, \sigma_n)$

$$w'_0(\sigma) = \frac{1}{N}\Big(\frac{\alpha}{N}\Big)^{s(\sigma)}\Big(1 - \alpha + \frac{\alpha}{N}\Big)^{n-s(\sigma)}$$

Marginalized weights

$$w'_0(\sigma_{1:t}) = \sum_{\sigma' \in B : \sigma'_{1:t} = \sigma_{1:t}} w'_0(\sigma')$$

# The Fixed-Share Forecaster (cont'd)

**Details**: everything starts from a non–uniform belief over the class $B$ of *all* the possible switching strategies $\sigma = (\sigma_1, \dots, \sigma_n)$

$$w'_0(\sigma) = \frac{1}{N}\left(\frac{\alpha}{N}\right)^{s(\sigma)}\left(1 - \alpha + \frac{\alpha}{N}\right)^{n-s(\sigma)}$$

Marginalized weights

$$w'_0(\sigma_{1:t}) = \sum_{\sigma' \in B : \sigma'_{1:t} = \sigma_{1:t}} w'_0(\sigma')$$

Recursive forumlation

$$w'_0(\sigma_1) = 1/N$$

$$w'_0(\sigma_{1:t+1}) = w'_0(\sigma_{1:t})\left(\frac{\alpha}{N} + (1-\alpha)\mathbb{I}\left\{\sigma_{t+1} = \sigma_t\right\}\right)$$

# The Fixed-Share Forecaster (cont'd)

The value

$$p = \frac{w_0'(\sigma_{1:t+1})}{w_0'(\sigma_{1:t})} = \frac{\alpha}{N} + (1-\alpha)\mathbb{I}\{\sigma_{t+1} = \sigma_t\}$$

is the conditional probability that a random sequence $(I_1, \dots, I_n)$ drawn from $w_0'$ has $I_{t+1} = \sigma_{t+1}$ given that $I_t = \sigma_t$

# The Fixed-Share Forecaster (cont'd)

The value

$$p = \frac{w_0'(\sigma_{1:t+1})}{w_0'(\sigma_{1:t})} = \frac{\alpha}{N} + (1-\alpha)\mathbb{I}\left\{\sigma_{t+1} = \sigma_t\right\}$$

is the conditional probability that a random sequence $(I_1, \ldots, I_n)$ drawn from $w_0'$ has $I_{t+1} = \sigma_{t+1}$ given that $I_t = \sigma_t$

Let $X = \{1, \ldots, N\}$ be the state of a Markov chain $M$

- $\mathbb{P}[X_1 = i] = w_0'(i_1) = 1/N$
- $\mathbb{P}[X_{t+1} = i | X_t = j] = \alpha/N$ (if $i \neq j$)
- $\mathbb{P}[X_{t+1} = i | X_t = i] = 1 - \alpha + \alpha/N$

$\Rightarrow$ The weights $w_0'$ encode a joint distribution of a Markov chain $M$ such that $X_1$ is drawn uniformly at random and $X_{t+1}$ is equal to the previous expert $X_t$ with probability $1-\alpha+\alpha/N$ and is equal to $j \neq X_t$ with probability $\alpha/N$.

# The Fixed-Share Forecaster (cont'd)

The value

$$p = \frac{w_0'(\sigma_{1:t+1})}{w_0'(\sigma_{1:t})} = \frac{\alpha}{N} + (1-\alpha)\mathbb{I}\{\sigma_{t+1} = \sigma_t\}$$

is the conditional probability that a random sequence $(I_1, \ldots, I_n)$ drawn from $w_0'$ has $I_{t+1} = \sigma_{t+1}$ given that $I_t = \sigma_t$

Let $X = \{1, \ldots, N\}$ be the state of a Markov chain $M$

- $\mathbb{P}[X_1 = i] = w_0'(i_1) = 1/N$
- $\mathbb{P}[X_{t+1} = i | X_t = j] = \alpha/N$ (if $i \neq j$)
- $\mathbb{P}[X_{t+1} = i | X_t = i] = 1 - \alpha + \alpha/N$

$\Rightarrow$ small $\alpha$ corresponds to small weight to switching experts with many switches

# The Fixed-Share Forecaster (cont'd)

At round $t$, the weight

$$w'_t(\sigma) = w'_0(\sigma) \exp\left(\eta \sum_{s=1}^{t} \ell(f_{\sigma_s,t}, y_s)\right)$$

is used to randomized over *switching experts* which reduces to a randomization over *base expert*

$$w'_{i,t} = \sum_{\sigma \in B : \sigma_t = i} w'_t(\sigma)$$

with $w'_{i,t} = 1/N$.

# The Fixed-Share Forecaster (cont'd)

## Theorem

*The Fixed-Share Forecaster with parameters* $\eta, \alpha$ *has a regret w.r.t. any switching expert* $\sigma$

$$R_n(\mathcal{A}) \leq \frac{s(\sigma)+1}{\eta} \log N + \frac{1}{\eta} \log \frac{1}{(\alpha/N)^{s(\sigma)}(1-\alpha)^{n-s(\sigma)-1}} + \frac{\eta}{8} n$$

# The Fixed-Share Forecaster (cont'd)

**Corollary**

*The Fixed-Share Forecaster with a suitable parameter $\eta$ and $\alpha = m/(n-1)$ has a regret w.r.t. any switching expert $\sigma$ with $s(\sigma) \leq m$*

$$R_n(\mathcal{A}) \leq \sqrt{\frac{8}{n}\Big((m+1)\log N + (n-1)H\big(\frac{m}{n-1}\big)\Big)}$$

# The Fixed-Share Forecaster (cont'd)

**Corollary**

*The Fixed-Share Forecaster with a suitable parameter $\eta$ and $\alpha = m/(n-1)$ has a regret w.r.t. any switching expert $\sigma$ with $s(\sigma) \leq m$*

$$R_n(\mathcal{A}) \leq \sqrt{\frac{8}{n}\Big((m+1)\log N + (n-1)H\big(\frac{m}{n-1}\big)\Big)}$$

**Remark**: $\alpha$ encodes the *frequency of switch* and it allows the algorithm to compete against $m \approx \alpha n$ switching experts.

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts
- Tracking the Best Expert
- Tree Experts
- Shortest Path Problem
- Infinite Experts

$$ How to Make Money with Online Learning $$

[Conc]lusions

# Tree Experts

Instead of *switching* experts we now consider *tree experts*.

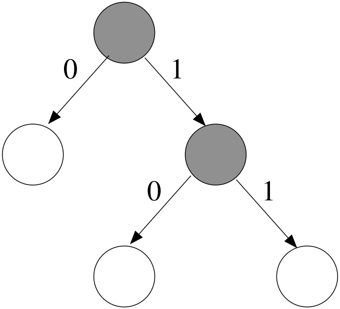
# Tree Experts

Instead of *switching* experts we now consider *tree experts*.

Let's consider the discrete binary prediction case $\mathcal{Y} = \{0, 1\}$.

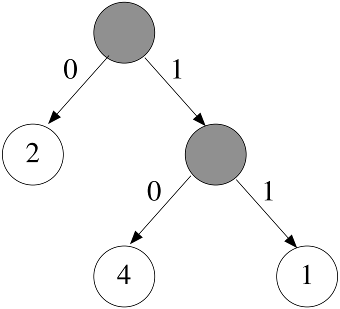
# Tree Experts (cont'd)

A binary tree



# Tree Experts (cont'd)

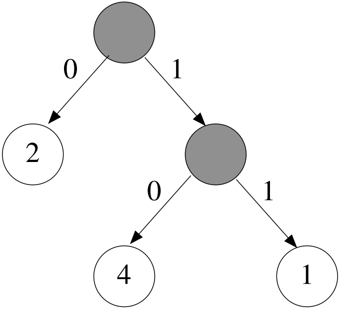
An expert tree



# Tree Experts (cont'd)

We traverse the tree according to the past observations (in reversed order)

$$(y_{t-1}, y_{t-2}, \ldots, y_{t-d})$$




See example on the board...

# Tree Experts (cont'd)

An expert tree $E$ has

- Number of leaves leaves$(E)$
- Number of nodes $||E||$
- $D$-size of an expert $||E||_D = ||E|| - |\{\text{leaves at depth} D\}|$

# Tree Experts (cont'd)

Inefficient EWA algorithm over experts

- Initial weights

$$w_{E,0} = 2^{-||E||_D} N^{-|\text{leaves}(E)|}$$

# Tree Experts (cont'd)

Inefficient EWA algorithm over experts

- Initial weights

$$w_{E,0} = 2^{-||E||_D} N^{-|\mathrm{leaves}(E)|}$$

- At round $t$

$$w_{E,t-1} = w_{E,0} \prod_{v \in \mathrm{leaves}(E)} w_{E,v,t-1}$$

# Tree Experts (cont'd)

Inefficient EWA algorithm over experts

- Initial weights

$$w_{E,0} = 2^{-||E||_D} N^{-|\text{leaves}(E)|}$$

- At round $t$

$$w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1}$$

- Leaf weight

$$w_{E,v,t} = \begin{cases} w_{E,v,t-1} \exp\big(-\eta\ell(f_{i_E(v),t}, y_t)\big) & \text{if } v \text{ is active} \\ w_{E,v,t-1} & \text{otherwise} \end{cases}$$

# Tree Experts (cont'd)

Inefficient EWA algorithm over experts

- Initial weights

$$w_{E,0} = 2^{-||E||_D} N^{-|\text{leaves}(E)|}$$

- At round $t$

$$w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1}$$

- Leaf weight

$$w_{E,v,t} = \begin{cases} w_{E,v,t-1} \exp\big( -\eta \ell(f_{i_E(v),t}, y_t)\big) & \text{if } v \text{ is active} \\ w_{E,v,t-1} & \text{otherwise} \end{cases}$$

- Randomize over

$$p_{i,t} = \frac{\sum_E \mathbb{I}\left\{i_E(\mathbf{y}^t) = i\right\} w_{E,t-1}}{\sum_{E'} w_{E',t-1}}$$

# Tree Experts (cont'd)

**Theorem**

*The randomized EWA($\eta$) over the set of experts of depth $D$ satisfies for any tree expert $E$*

$$R_n \leq \frac{||E||_D}{\eta} \log 2 + \frac{|leaves(E)|}{\eta} \log N + \frac{\eta}{8} n$$

*if $\eta$ is optimized*

$$R_n \leq \sqrt{n 2^{D-1} \log(2N)}$$

# Tree Experts (cont'd)

**Theorem**

*The randomized EWA($\eta$) over the set of experts of depth $D$ satisfies for any tree expert $E$*

$$R_n \leq \frac{||E||_D}{\eta} \log 2 + \frac{|leaves(E)|}{\eta} \log N + \frac{\eta}{8} n$$

*if $\eta$ is optimized*

$$R_n \leq \sqrt{n 2^{D-1} \log(2N)}$$

**Problem**: again, the number of experts of $D$ maybe very large and the number of leaves even larger, so this algorithm is *infeasible*

# Tree Experts (cont'd)

There exists an efficient *tree expert forecaster* with $N(2^{D+1}-1)$ weights, which is $N$ weights for each node of the complete binary tree of depth $D$.

No details here but the algorithm involves a *recursive* update of the weights of the nodes.

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

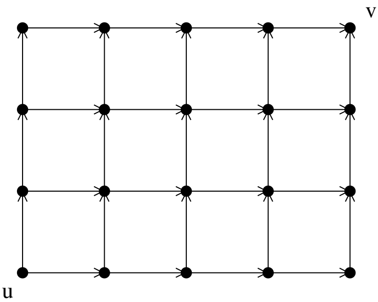
Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts
- Tracking the Best Expert
- Tree Experts
- Shortest Path Problem
- Infinite Experts

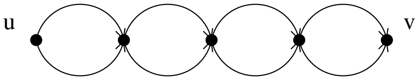
$$ How to Make Money with Online Learning $$

…clusions

# Directed Acyclic Graphs



# Directed Acyclic Graphs



# Directed Acyclic Graphs (cont'd)

A directed acyclic graph is

- set of edges $E = \{e_1, \ldots, e_{|E|}\}$
- set of vertices $V$
- $\Rightarrow e = (v_1, v_2)$

Paths

- Start vertex $u$, end vertex $v$
- Path from $u$ to $v$ is $e^{(1)}, \ldots, e^{(k)}$ with $e^{(1)} = (u, v_1)$, $e^{(j)} = (v_{j-1}, v_j)$
- Path $\mathbf{i} \in \{0, 1\}^{|E|}$

# Directed Acyclic Graphs (cont'd)

At each round $t$

- each edge $e_j$ has a loss $\ell_{e_j,t}$
- the whole graph has $y_t = \ell_t \in [0,1]^{|E|}$
- the loss of a path $\mathbf{i}$ is $\ell(\mathbf{i}, y_t) = \mathbf{i} \cdot \ell_t = \sum_j \ell_{e_j,t} \mathbb{I}\{i_j = 1\}$

# Directed Acyclic Graphs (cont'd)

At each round $t$

- each edge $e_j$ has a loss $\ell_{e_j,t}$
- the whole graph has $y_t = \ell_t \in [0,1]^{|E|}$
- the loss of a path $\mathbf{i}$ is $\ell(\mathbf{i}, y_t) = \mathbf{i} \cdot \ell_t = \sum_j \ell_{e_j,t} \mathbb{I}\{i_j = 1\}$

Regret

$$R_n(\mathcal{A}) = \sum_{t=1}^{n} \mathbb{E}[\ell(\mathbf{I}_t, Y_t)] - \min_{\mathbf{i}} \sum_{t=1}^{n} \ell(\mathbf{i}_t, Y_t)$$

# Follow the Perturbed Leader

At round $t$ the leader is

$$\arg\min_{\mathbf{i}} \mathbf{i} \cdot \Big( \sum_{s=1}^{t-1} \ell_s \Big)$$

# Follow the Perturbed Leader

At round $t$ the leader is

$$\arg\min_{\mathbf{i}} \mathbf{i} \cdot \Big( \sum_{s=1}^{t-1} \ell_s \Big)$$

Let $\mathbf{Z}_t \in \mathbb{R}^{|E|}$ be a random variable.
The perturbed leader is

$$I_t = \arg\min_{\mathbf{i}} \mathbf{i} \cdot \Big( \sum_{s=1}^{t-1} \ell_s + Z_t \Big)$$

# Follow the Perturbed Leader (cont'd)

The perturbed leader is

$$I_t = \arg\min_{\mathbf{i}} \mathbf{i} \cdot \Big( \sum_{s=1}^{t-1} \ell_s + Z_t \Big)$$

There exist efficient algorithms to find the *shortest path* in a directed acyclic graph in *linear time*.

# Follow the Perturbed Leader (cont'd)

## Theorem

*Consider the follow-the-perturbed-leader with noise vectors $Z_t \in [0, \Delta]^{|E|}$. Then with probability $1 - \delta$*

$$R_n \leq K\Delta + \frac{nK|E|}{\Delta} + K\sqrt{\frac{n}{2}\log\frac{1}{\delta}}$$

*with $K$ the length of the longest path from $u$ to $v$.*
*By setting $\Delta = \sqrt{n|E|}$ we have*

$$R_n \leq 2K\sqrt{n|E|} + K\sqrt{n/2\log(1/\delta)}$$

# Exponentially Weighted Average for Graphs

**Infeasible solution**: simply list all the possible paths and consider them as experts

**Efficient solution**: build the predicted path $\mathbf{I}_t$ by selecting edges one by one

# Exponentially Weighted Average for Graphs

Edge cumulative loss

$$L_{e,t} = \sum_{s=1}^{t} \ell_{e,s}$$

Let $\mathcal{P}_w$ the set of paths from vertex $w \in V$ to end vertex $v$, we define

$$G_t(w) = \sum_{\mathbf{i} \in \mathcal{P}_w} \exp\Big( -\eta \sum_{e \in \mathbf{i}} L_{e,t} \Big)$$

# Exponentially Weighted Average for Graphs

We order the vertices as $v_1, \ldots, v_{|V|}$ so that

$$u = v_1, v = v_{|V|}$$

and if $i < j$ then there is no edge between $v_i$ and $v_j$ (exploiting the structure of the directed acyclic graph).

# Exponentially Weighted Average for Graphs

Given the ordering, we can computed $G_t(w)$ recursively

$$G_t(v) = 1$$

If $G_t(v_i)$ has been calculated for all $v_i$ with $i = |V|, |V-1|, \ldots, j+1$, then

$$G_t(v_j) = \sum_{w:(v_j,w)\in E} G_t(w) \exp\big(-\eta L_{(v_j,w),t}\big)$$

# Exponentially Weighted Average for Graphs

From the weights on the edge to the (random) path $\mathbf{I}_t$.
Start from $u$, then for any $k = 1, \ldots$

- Pick the vertex $v_{I_t,k}$ with probability

$$\mathbb{P}[v_{\mathbf{I}_t,k} = v_{\mathbf{i},k} | v_{\mathbf{I}_t,k-1} = v_{\mathbf{i},k-1}, \ldots, v_{\mathbf{I}_t,0} = v_{\mathbf{i},0}]$$
$$= \begin{cases} \frac{G_{t-1}(v_{\mathbf{i},k})}{G_{t-1}(v_{\mathbf{i},k-1})} & \text{if } (v_{\mathbf{i},k-1}, v_{\mathbf{i},j}) \in E \\ 0 & \text{otherwise.} \end{cases}$$

# Exponentially Weighted Average for Graphs

**Theorem**

*The efficient EWA achieves a regret*

$$R_n \leq K\left(\frac{\log M}{\eta} + \frac{n\eta}{8} + \sqrt{\frac{n}{2}\log\frac{1}{\delta}}\right)$$

*with probability $1-\delta$, where $M$ is the total number of paths from $u$ to $v$ and $K$ is the length of the longest path.*

# Exponentially Weighted Average for Graphs

## Theorem

*The efficient EWA achieves a regret*

$$R_n \leq K\left(\frac{\log M}{\eta} + \frac{n\eta}{8} + \sqrt{\frac{n}{2}\log\frac{1}{\delta}}\right)$$

*with probability* $1-\delta$*, where* $M$ *is the total number of paths from* $u$ *to* $v$ *and* $K$ *is the length of the longest path.*

**Comparison**: the performance is much better than the perturbed leader ($O(\sqrt{n|E|})$).

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts
- Tracking the Best Expert
- Tree Experts
- Shortest Path Problem
- Infinite Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Infinite Experts: Sequential Investment

**Problem**: the bounds displays a nice dependency $\log N$, but what if the number of experts is infinite?

# Infinite Experts: Sequential Investment (cont'd)

An example in *sequential investment* (portfolio selection)

- $d$ stocks
- market vector $z \in \mathbb{R}_+^d$
- portfolio allocation $a \in \Delta^d$ (i.e., $a_i \in [0,1]$ and $\sum_{i=1}^d a_i = 1$)
- the capital $W$ evolves as

$$W_t = \sum_{i=1}^{d} \underbrace{a_t(i) W_{t-1}}_{\text{fraction on stock } i} z_t(i) = W_{t-1} a_t^\top z_t = W_0 \prod_{s=1}^{t} a_s^\top z_s$$

# Infinite Experts: Sequential Investment (cont'd)

The prediction game

- Experts: all the *constantly rebalanced portfolios* (i.e., constant portfolio $a$ over $n$ rounds)
- Expert performance $W_n(a) = W_0 \prod_{t=1}^{n} a^\top z_t$
- *Best expert* $\sup_{a \in \Delta^d} W_n(a)$
- Performance of $\mathcal{A}$ (sequence of portfolios $a_1, \ldots, a_n$):

$$\text{Competitive wealth ratio: } \frac{\sup_a W_n(a)}{W_n(\mathcal{A})}$$

# Infinite Experts: Sequential Investment (cont'd)

The prediction game

- Experts: all the *constantly rebalanced portfolios* (i.e., constant portfolio $a$ over $n$ rounds)
- Expert performance $W_n(a) = W_0 \prod_{t=1}^n a^\top z_t$
- *Best expert* $\sup_{a\in\Delta^d} W_n(a)$
- Performance of $\mathcal{A}$ (sequence of portfolios $a_1, \ldots, a_n$):

$$\text{Log wealth ratio: } \log\left(\frac{\sup_a W_n(a)}{W_n(\mathcal{A})}\right)$$

# Infinite Experts: Sequential Investment (cont'd)

The prediction game

- Experts: all the *constantly rebalanced portfolios* (i.e., constant portfolio $a$ over $n$ rounds)
- Expert performance $W_n(a) = W_0 \prod_{t=1}^{n} a^\top z_t$
- *Best expert* $\sup_{a \in \Delta^d} W_n(a)$
- Performance of $\mathcal{A}$ (sequence of portfolios $a_1, \ldots, a_n$):

Log wealth ratio: $$\sum_{t=1}^{n} -\log(a_t^\top z_t) - \inf_{a \in \Delta^d} \sum_{t=1}^{n} -\log(a^\top z_t)$$

# Infinite Experts: Sequential Investment (cont'd)

The prediction game

- Experts: all the *constantly rebalanced portfolios* (i.e., constant portfolio $a$ over $n$ rounds)
- Expert performance $W_n(a) = W_0 \prod_{t=1}^{n} a^\top z_t$
- *Best expert* $\sup_{a \in \Delta^d} W_n(a)$
- Performance of $\mathcal{A}$ (sequence of portfolios $a_1, \ldots, a_n$):

$$\text{Regret: } \sum_{t=1}^{n} \ell(a_t, z_t) - \inf_{a \in \Delta^d} \sum_{t=1}^{n} \ell(a, z_t)$$

# Infinite Experts: Sequential Investment (cont'd)

Continuous EWA($\eta$)
At each round $t$, switch to position

$$a_t = \int_{a \in \Delta^d} \frac{w_t(a)}{W_t} a \, da$$

with

$$w_t(a) = \exp\Big( -\eta \sum_{s=1}^{t-1} \ell(a, z_s) \Big), \quad W_t = \int_a w_t(a) \, da$$

# Infinite Experts: Sequential Investment (cont'd)

**Problem**: the portfolio selection

$$a_t = \int_{a \in \Delta^d} \frac{w_t(a)}{W_t} a \, da$$

is easy to write but how easy is it to *compute*?

# Infinite Experts: Sequential Investment (cont'd)

**Problem**: the portfolio selection

$$a_t = \int_{a \in \Delta^d} \frac{w_t(a)}{W_t} a \, da$$

is easy to write but how easy is it to *compute*?
*Easy!* (or at least not too much complicated...)

# Infinite Experts: Sequential Investment (cont'd)

**Remark**: notice that

$$a_t = \int_{a \in \Delta^d} \frac{w_t(a)}{W_t} a \, da$$

is an integration problem with a measure $w_t(a)/W_t$ and that

$$f_t(a) : a \mapsto \frac{w_t(a)}{W_t} = \frac{1}{W_t} \exp\Big( -\eta \sum_{s=1}^{t-1} \ell(a, z_s) \Big)$$

is a log–concave function and $\Delta_d$ is a convex set

# Infinite Experts: Sequential Investment (cont'd)

**Remark**: notice that

$$a_t = \int_{a \in \Delta^d} \frac{w_t(a)}{W_t} a \, da$$

is an integration problem with a measure $w_t(a)/W_t$ and that

$$f_t(a) : a \mapsto \frac{w_t(a)}{W_t} = \frac{1}{W_t} \exp\Big( -\eta \sum_{s=1}^{t-1} \ell(a, z_s) \Big)$$

is a log–concave function and $\Delta_d$ is a convex set

$\Rightarrow$ we can use random walk methods which are particularly efficient

# Infinite Experts: Sequential Investment (cont'd)

*A sketch of the algorithm*

**Input**: $m$, $\sigma$
Average over $m$ samples obtained as

- Start from a uniform allocation $a_0 = (1/d, \ldots, 1/d)$
- Repeat for $T$ steps
  - Choose a dimension $j$ (i.e., a stock) at random
  - Choose a value $X \in \{-1, 1\}$ at random
    - Compute $p_1 = f(a)$
    - Compute $p_2 = f(a(1), \ldots, a(j) + X\sigma, \ldots, a(d) - X\sigma)$
    - With probability $p_1/p_2$ update $a(j) = a(j) + \sigma X$ and $a(d) = a(d) - \sigma X$

# Infinite Experts: Sequential Investment (cont'd)

**Theorem**

*If*

$$m \geq O\Big(\frac{n^3}{\epsilon^2}\log\frac{dn}{\delta}\Big)$$

$$S \geq O\Big(\frac{d}{\sigma^2}\log\frac{d}{\epsilon\sigma}\Big)$$

*then random walk algorithm performs* $(1-\epsilon)$ *times as well as the exact algorithm with probability* $1-\delta$*.*

# Extension to Infinite Experts

**Theorem**

*Given a convex loss bounded in $[0,1]$, for any $\gamma > 0$, the (exact) Continuous EWA($\eta$) achieves a regret*

$$R_n \leq \frac{d \log \frac{1}{\gamma}}{\eta} + \frac{n\eta}{8} + \gamma n$$

*By setting $\gamma = 1/n$ and $\eta = 2\sqrt{2d \log n/n}$ then*

$$R_n \leq 1 + \sqrt{\frac{dn \log n}{2}}$$

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# The Betting Problem

## Disclaimer

***Neither the authors nor the lecturer are responsible for any inappropriate use of the techniques presented in this course.***

# The Betting Problem

**The problem**: Predict the outcome of a game using the odds from the bookmakers.

# Glossary

- *Bookmaker (bookie)*: The company organizing the gambling
- *Odds*: Bookmaker's view of the chance of a competitor winning (adjusted to include a profit).
- *Stake*: The money you bet.
- *Overround*: Profit margin in the bookmaker's favor.

# Glossary (cont'd)

*Theoretical* (in favor) odds

- **Example**: There are 5 pink marbles, 2 blue marbles, and 8 purple marbles. What are the odds in favor of picking one blue marble?

# Glossary (cont'd)

*Theoretical* (in favor) odds

- **Example**: There are 5 pink marbles, 2 blue marbles, and 8 purple marbles. What are the odds in favor of picking one blue marble?
  **Answer**: 2/13 (2:13)

# Glossary (cont'd)

*Theoretical* (in favor) odds

- **Example**: There are 5 pink marbles, 2 blue marbles, and 8 purple marbles. What are the odds in favor of picking one blue marble?
  **Answer**: 2/13 (2:13)
- **Definition**:

$$\text{odd} = \frac{\text{prob. in favor}}{\text{prob. against}}$$

Source: wikipedia

# Glossary (cont'd)

*Theoretical* (in favor) odds

- **Example**: There are 5 pink marbles, 2 blue marbles, and 8 purple marbles. What are the odds in favor of picking one blue marble?
  **Answer**: 2/13 (2:13)
- **Definition**:

$$a = \frac{p}{1-p}$$

Source: wikipedia

# Glossary (cont'd)

*Theoretical* (in favor) odds

- **Example**: There are 5 pink marbles, 2 blue marbles, and 8 purple marbles. What are the odds in favor of picking one blue marble?
  **Answer**: 2/13 (2:13)
- **Definition**:

$$a = \frac{p}{1 - p}$$

  If $p = 0.2$, the odds are $a = 0.25$, and represent the stake necessary to *win one unit (plus the bet) on a successful wager* when offered fair odds.
- Odds $a = 0.25$ correspond to *fractional odds* are 4 to 1 (4:1), in *decimal odds* are 5.0.

Source: wikipedia

# Glossary (cont'd)

*Theoretical* (against) odds

$$a = \frac{1-p}{p}$$

# Glossary (cont'd)

*Theoretical* (against) odds

$$a = \frac{1-p}{p}$$

In the previous example: What are the odds *against* picking one blue marble? 13 : 2

# Glossary (cont'd)

*Gambling* odds

- Bookmaker's odds include a profit margin, the *over-round*.
- **Example**: In a 3-horse race, let 50%, 40% and 10% be the *true* probabilities (odds 5-5, 6-4 and 9-1). The bookmaker may increase the values to 60%, 50% and 20% (odds 4-6, 5-5 and 4-1). These values total 130, meaning that the book has an *overround of 30*.

# Glossary (cont'd)

From odds to probabilities:

- $K$ possible outcomes
- $K$ odds $a_1, \ldots, a_K$
- Probabilities

$$p_k = \frac{1/a_k}{\sum_{k'=1}^{K} 1/a_{k'}}$$

# The Brier's Game

- Outcome space: *possible results*
- Decision space: *probability distribution*
- Set of experts: *bookmakers*
- Loss function: *quadratic loss on the probability distribution*

# The Brier's Game

- Outcome space: $\mathcal{Y} = \{1, \ldots, K\}$
- Decision space: $\mathcal{D} = \mathbb{P}(\mathcal{Y})$
- Set of experts: $1, \ldots, N$
- Loss function:

$$\ell(y, \hat{\mathbf{p}}) = \sum_{k=1}^{K} (\hat{p}(k) - \delta_y(k))^2$$

# The Brier's Game

At each round $t$

- Expert $i$ predicts a distribution over outcomes $\mathbf{p}_{i,t}$

# The Brier's Game

At each round $t$

- Expert $i$ predicts a distribution over outcomes $\mathbf{p}_{i,t}$
- Learner predicts a distribution over outcomes $\hat{\mathbf{p}}_t$

# The Brier's Game

At each round $t$

- Expert $i$ predicts a distribution over outcomes $\mathbf{p}_{i,t}$
- Learner predicts a distribution over outcomes $\hat{\mathbf{p}}_t$
- Reality announces the outcome $y_t$

# The Brier's Game

At each round $t$

- Expert $i$ predicts a distribution over outcomes $\mathbf{p}_{i,t}$
- Learner predicts a distribution over outcomes $\hat{\mathbf{p}}_t$
- Reality announces the outcome $y_t$
- Learner incurs a loss $\ell(y_t, \hat{\mathbf{p}}_t)$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$
- Compute

$$G_t(y) = -\log\Big(\sum_{i=1}^{N} w_{i,t-1}\exp(-\ell(y, \mathbf{p}_{i,t}))\Big)$$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$
- Compute

$$G_t(y) = -\log\Big(\sum_{i=1}^{N} w_{i,t-1}\exp(-\ell(y,\mathbf{p}_{i,t}))\Big)$$

- Solve $\sum_y (s - G_t(y))^+ = 2$ with $s \in \mathbb{R}$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$
- Compute

$$G_t(y) = -\log\Big(\sum_{i=1}^{N} w_{i,t-1} \exp(-\ell(y, \mathbf{p}_{i,t}))\Big)$$

- Solve $\sum_y (s - G_t(y))^+ = 2$ with $s \in \mathbb{R}$
- Set $\hat{p}_t(k) = (s - G_t(k))^+/2$ for any $k \in \mathcal{Y}$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$
- Compute

$$G_t(y) = -\log\left(\sum_{i=1}^{N} w_{i,t-1} \exp(-\ell(y, \mathbf{p}_{i,t}))\right)$$

- Solve $\sum_y (s - G_t(y))^+ = 2$ with $s \in \mathbb{R}$
- Set $\hat{p}_t(k) = (s - G_t(k))^+/2$ for any $k \in \mathcal{Y}$
- Predict $\hat{\mathbf{p}}_t$ and observe $y_t$

# Strong Aggregating Algorithm

Initialize the weights $w_{i,0} = 1$

- Record the experts' predictions $\mathbf{p}_{i,t}$
- Compute

$$G_t(y) = -\log\Big(\sum_{i=1}^{N} w_{i,t-1} \exp(-\ell(y, \mathbf{p}_{i,t}))\Big)$$

- Solve $\sum_y (s - G_t(y))^+ = 2$ with $s \in \mathbb{R}$
- Set $\hat{p}_t(k) = (s - G_t(k))^+/2$ for any $k \in \mathcal{Y}$
- Predict $\hat{\mathbf{p}}_t$ and observe $y_t$
- Update $w_{i,t} = w_{i,t-1} \exp(-\ell(y, \mathbf{p}_{i,t}))$

# Strong Aggregating Algorithm

A *rough* explanation

- $\exp(-\ell(y, \mathbf{p}_{i,t}))$ is the "loss" suffered by $i$ if the outcome will be $y$
- $G_t(y)$ is a mixing function of the the *potential* losses using weights $w$s
- We search for a mapping function $\Sigma$ which takes $G$ and returns *valid* predictions such that

$$\ell(y, \Sigma(G)) \leq G(y)$$

# Strong Aggregating Algorithm

**Theorem**

*The strong aggregating algorithm on the Brier's game achieves a cumulative loss*

$$L_n(\mathcal{A}) \leq \min_{1 \leq i \leq N} L_{i,n} + \log N$$

# Strong Aggregating Algorithm

### Theorem

*The strong aggregating algorithm on the Brier's game achieves a cumulative loss*

$$L_n(\mathcal{A}) \leq \min_{1 \leq i \leq N} L_{i,n} + \log N$$

**Remark**: and *no algorithm* can do better!

# Empirical Results

Available at: http://vovk.net/ICML2008/

# Empirical Results

Available at: http://vovk.net/ICML2008/
Database football

- 8999 matches in English football competitions over 4 years
- Outcomes: {home win, draw, away win}
- 8 Bookmakers (Bet365, Bet&Win, ...)

# Empirical Results

Available at: http://vovk.net/ICML2008/

Database football

- 8999 matches in English football competitions over 4 years
- Outcomes: {home win, draw, away win}
- 8 Bookmakers (Bet365, Bet&Win, ...)

Database tennis

- 10,087 matches in different tournaments over 4 years
- Outcomes: {player1 win, player2 win}
- 4 Bookmakers (Bet365, Bet&Win, ...)

# Empirical Results

Available at: http://vovk.net/ICML2008/

Database football

- 8999 matches in English football competitions over 4 years
- Outcomes: {home win, draw, away win}
- 8 Bookmakers (Bet365, Bet&Win, ...)

Database tennis

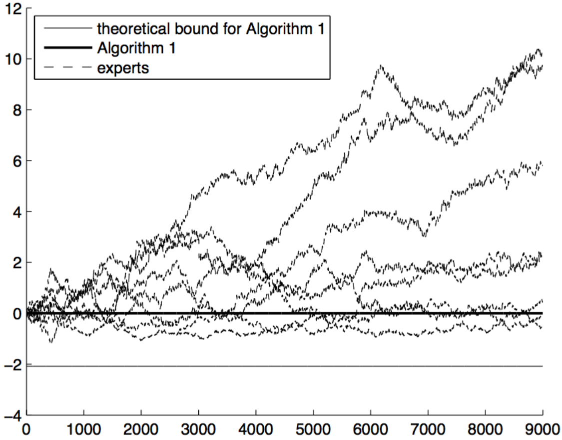
- 10,087 matches in different tournaments over 4 years
- Outcomes: {player1 win, player2 win}
- 4 Bookmakers (Bet365, Bet&Win, ...)

Pre-processing: from odds to probabilities

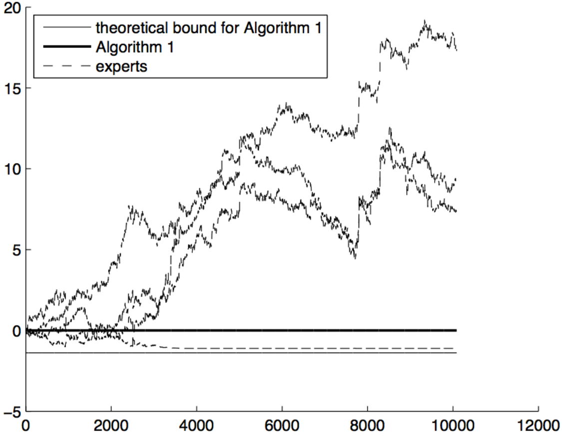
$$p(k) = a(k)^{-\gamma}$$

where $\gamma$ is related to the overround.

# Empirical Results: football



# Empirical Results: tennis



# Empirical Results: comparisons

**Question**: Independently from the theory is the SAA really good compared to other algorithms?

# Empirical Results: comparisons

**Question**: Independently from the theory is the SAA really good compared to other algorithms?

- Weighted average: the same as SSA but no function $G$
- Hedge (EWA)
- Weak aggregating

# Empirical Results: comparisons

Football results

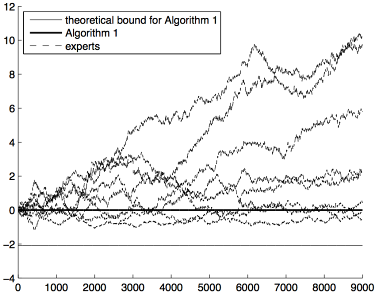
| *Algorithm* | *Maximal Difference* | *Theoretical Bound* |
|---|---|---|
| *Aggregating* | *1.1562* | *2.0794* |
| Weighted Average | *1.8697* | 16.6355 |
| Hedge | 4.5662 | 234.1159 |
| Weak Aggregating | 2.4755 | 464.0728 |

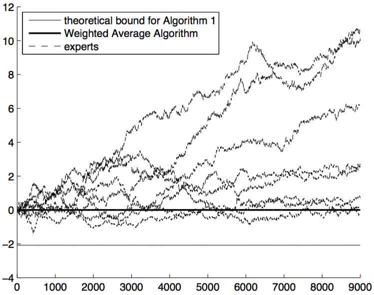
# Empirical Results: comparisons

Tennis results

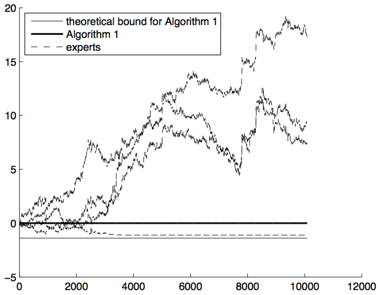
| *Algorithm* | *Maximal Difference* | *Theoretical Bound* |
|---|---|---|
| *Aggregating* | *1.2021* | *1.3863* |
| Weighted Average | *3.0566* | 11.0904 |
| Hedge | 9.0598 | 237.8904 |
| Weak Aggregating | 3.6101 | 473.0083 |

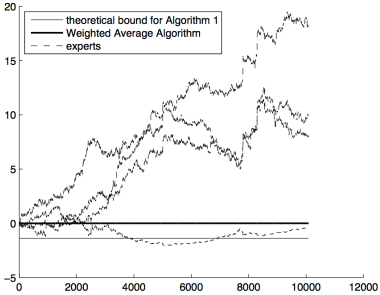
# Empirical Results: comparisons



# Empirical Results: comparisons



# Empirical Results: comparisons

Other observations

- SAA is able to (explicitly) *exploit* the shape of the *loss function*
- Other algorithms are *less aware* of the loss function
- Experiments (not reported) on other algorithms, show that non-theoretically guaranteed algorithms *do not perform that poorly* but are much *less robust*

# Discussion

- Is it possible to add side information?
- Is it the minimization of the regret wrt the best expert our real goal?
- Is it possible to merge *model-based* prediction and *expert-based* prediction?

# Outline

Introduction

Continuous Prediction with Expert Advice: the EWA

Discrete Prediction with Expert Advice: the EWA

Efficient Forecasters for Large Classes of Experts

$$ How to Make Money with Online Learning $$

Conclusions

# Other Online Learning Algorithms

- Follow-the-regularized leader
- The perceptron
- Proximal point algorithm
- Exponentiated gradient algorithms
- Mirror decent
- Passive-agressive algorithm
- ...

# Other Online Learning Settings

- Online learning with partial monitoring
- Label-efficient learning
- Online learning in games
- Online binary classification
- Specific losses
- Contextual learning
- Hybrid stochastic-adversarial models
- ...

# Applications of Online Learning

- Stock market prediction (universal portfolio)
- Betting strategies
- Ozone ensamble prediction
- Online email categorization
- Speech-to-text and Music-to-score Alignement
- ...

# Things to Remember

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem
- Online learning is about algorithms which are *robust* enough to working well in *any case*

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem
- Online learning is about algorithms which are *robust* enough to working well in *any case*
- In the expert advice model we can leverage on *many experts of any kind*

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem
- Online learning is about algorithms which are *robust* enough to working well in *any case*
- In the expert advice model we can leverage on *many experts of any kind*
- The *EWA* is a very flexible algorithm for both continuous and discrete prediction

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem
- Online learning is about algorithms which are *robust* enough to working well in *any case*
- In the expert advice model we can leverage on *many experts of any kind*
- The *EWA* is a very flexible algorithm for both continuous and discrete prediction
- Theory gives you *worst-case* guarantees on the algorithm performance

# Things to Remember

- Learning when *data* are coming *in a stream* is a very relevant problem
- Online learning is about algorithms which are *robust* enough to working well in *any case*
- In the expert advice model we can leverage on *many experts of any kind*
- The *EWA* is a very flexible algorithm for both continuous and discrete prediction
- Theory gives you *worst-case* guarantees on the algorithm performance
- Many potential applications and *it works*

Advanced Topics in Machine Learning

# Part II: An Introduction to Online Learning

*Alessandro Lazaric*
alessandro.lazaric@inria.fr
sequel.lille.inria.fr