Machine Learning Theory. Lecture 6.

Wouter M. Koolen

- Online Learning, basic algorithms
- Specialists

Online learning focus

- Tight feedback loop (recurring prediction task)
- Continuous learning (no training/learning separation)
- Adversarial analysis (Prequential principle, individual sequence. There is only the data. Also establishes robustness of statistical estimators.)
- Emphasis on both computational and statistical performance
- Regret: relative notion of performance

Application domains

Truly sequential problems:

- electricity demand prediction (EDF, also Amazon)
- mobile device power management
- hybrid cars engine switching
- caching
- medical trials (bandits)
- online advertisement (bandits)
- weather forecasting
- data compression (CTW)
- statistical testing
- investment (Universal portfolios)
- input assistants (e.g. Dasher)
- prediction with expert advice (meld human and machine prediction)
- online convex optimisation

Wider application

- Big data sets (transport online algorithm state, online to batch conversion)
- Convex optimisation
- Game theory (online learning methods for approximate equilibrium)
- General understanding
  - Uncertainty and ways to manipulate it
  - Makeup of and patterns in data
  - Complexity of classes of strategies

The menu for today

Two fundamental and prototypical online learning problems

- The mix-loss game
  - Aggregating Algorithm
- The dot-loss game
  - Hedge
  - ML-Prod
- Specialists
  - Specialists Aggregating Algorithm

Mix-loss game

Protocol:

- For $t = 1, 2, \ldots$
  - Learner chooses a distribution $\boldsymbol{w}_t$ on K "experts".
  - Adversary reveals loss vector $\boldsymbol{\ell}_t \in (-\infty, \infty]^K$.
  - Learner's loss is the **mix loss** $-\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\ell_{t,k}}\right)$

Instances:

- Investment (loss is *negative log-growth*)
- Data compression (loss is *code length*)
- Probability forecasting (loss is *logarithmic loss*)

Mix-loss objective

Obviously we cannot guarantee small loss.

Idea: relative evaluation, i.e. performance close to best expert.

Definition: After T rounds of the mix-loss game, the *regret* is given by

$$R_T = \underbrace{\sum_{t=1}^{T} -\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\ell_{t,k}}\right)}_{\text{Learner's mix loss}} - \underbrace{\min_k \sum_{t=1}^{T} \ell_{t,k}}_{\text{loss of best expert}}$$

Goal: design an algorithm for Learner that guarantees low regret.

Mix-loss regret: lower bound (adversary)

Theorem: Any algorithm for Learner can be forced to incur regret $R_T \geq \ln K$, *already in* $T = 1$ *round*.

Idea: Look at $k_{\text{low}} = \arg\min_k w_{1,k}$ so that $w_{1,k_{\text{low}}} \leq \frac{1}{K}$.

Administer loss killing everyone but k_{low}

$$\ell_{1,k} = \begin{cases} \infty & k \neq k_{\text{low}} \\ 0 & k = k_{\text{low}} \end{cases}$$

Now Learner's mix loss equals

$$-\ln\left(\sum_{k=1}^{K} w_{1,k} e^{-\ell_{1,k}}\right) = -\ln\left(w_{1,k_{\text{low}}} e^{-\ell_{t,k_{\text{low}}}}\right) \geq \ln K + \ell_{t,k_{\text{low}}}$$

The Aggregating Algorithm for mix loss

Definition: The *Aggregating Algorithm* plays weights in round t:

$$w_{t,k} = \frac{e^{-\sum_{s=1}^{t-1} \ell_{s,k}}}{\sum_{j=1}^{K} e^{-\sum_{s=1}^{t-1} \ell_{s,j}}} \qquad \text{(AA)}$$

or, equivalently, $w_{1,k} = \frac{1}{K}$ and

$$w_{t+1,k} = \frac{w_{t,k} e^{-\ell_{t,k}}}{\sum_{j=1}^{K} w_{t,j} e^{-\ell_{t,j}}} \qquad \text{(AA, incremental)}$$

Many names

- (Generalisation of) Bayes rule
- Exponentially weighted average

Mix-loss regret: upper bound (algorithm)

Theorem: The regret of the Aggregating Algorithm does not exceed $R_T \le \ln K$ for all $T \ge 0$.

Proof: Crucial observation is that mix loss *telescopes*

$$\begin{aligned}
\sum_{t=1}^{T} -\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\ell_{t,k}}\right) &= \sum_{t=1}^{T} -\ln\left(\sum_{k=1}^{K} \frac{e^{-\sum_{s=1}^{t-1}\ell_{s,k}}}{\sum_{j=1}^{K} e^{-\sum_{s=1}^{t-1}\ell_{s,j}}} e^{-\ell_{t,k}}\right) \\
&= \sum_{t=1}^{T} -\ln\left(\frac{\sum_{k=1}^{K} e^{-\sum_{s=1}^{t}\ell_{s,k}}}{\sum_{j=1}^{K} e^{-\sum_{s=1}^{t-1}\ell_{s,j}}}\right) \\
&= -\ln\left(\sum_{k=1}^{K} e^{-\sum_{t=1}^{T}\ell_{t,k}}\right) + \ln K
\end{aligned}$$

and is bounded for each k by

$$\leq \sum_{t=1}^{T} \ell_{t,k} + \ln K \tag{1}$$

Dot-loss game

Protocol:

- For $t = 1, 2, \ldots$
  - Learner chooses a distribution $\boldsymbol{w}_t$ on K "experts".
  - Adversary reveals loss vector $\boldsymbol{\ell}_t \in [0, 1]^K$.
  - Learner's loss is the **dot loss** $\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t$

Many names:

- Decision Theoretic Online Learning
- Prediction with Expert Advice
- The Hedge setting

Dot-loss objective

Definition: *Regret* after T rounds:

$$R_T = \sum_{t=1}^{T} \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \min_k \sum_{t=1}^{T} \ell_{t,k}$$

Goal: design an algorithm for Learner that guarantees low regret.

Hedge algorithm

Idea: re-use AA for mix loss, now with learning rate η.

Definition: The *Hedge algorithm* with *learning rate* η plays weights in round t:

$$w_{t,k} = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_{s,k}}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_{s,j}}}. \qquad \text{(Hedge)}$$

or, equivalently, $w_{1,k} = \frac{1}{K}$ and

$$w_{t+1,k} = \frac{w_{t,k} e^{-\eta \ell_{t,k}}}{\sum_{j=1}^{K} w_{t,j} e^{-\eta \ell_{t,j}}} \qquad \text{(Hedge, incremental)}$$

Hedge analysis

Lemma: The regret of Hedge is bounded by

$$R_T \;\leq\; T\frac{\eta}{8} + \frac{\ln K}{\eta}$$

Proof:

$$\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t \;=\; \underbrace{\frac{-1}{\eta}\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\eta \ell_{t,k}}\right)}_{\text{mix loss}} + \underbrace{\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \frac{-1}{\eta}\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\eta \ell_{t,k}}\right)}_{\text{mixability gap}}$$

The mix loss telescopes, and is bounded by (1) by

$$\sum_{t=1}^{T} \frac{-1}{\eta}\ln\left(\sum_{k=1}^{K} w_{t,k} e^{-\eta \ell_{t,k}}\right) \;\leq\; \sum_{t=1}^{T} \ell_{t,k} + \frac{\ln K}{\eta}. \qquad (2)$$

The mixability gap is bounded by Hoeffding's Lemma (recall $\ell_{t,k} \in [0,1]$) by

$$\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} w_{t,k} e^{-\eta \ell_{t,k}} \right) \;\leq\; \frac{\eta}{8} \tag{3}$$

And over T rounds this accumulates to $T\frac{\eta}{8}$.

Putting (2) and (3) together yields the desired result.

Hedge tuning

Theorem: The Hedge regret bound is minimised at $\eta = \sqrt{\frac{8 \ln K}{T}}$ where it states

$$R_T \leq \sqrt{T/2 \ln K}.$$

ML-Prod

Second algorithm for dot loss game.

Advantages: better suited for powerful adaptivity (see homework).

ML-Prod for dot loss

Let's denote by $r_t^k = \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \ell_{t,k}$ the regret w.r.t. expert k in round t.

Definition: *ML-Prod* plays weights in round t:

$$w_{t,k} = \frac{\prod_{s=1}^{t-1}(1+\eta r_s^k)}{\sum_{j=1}^{K}\prod_{s=1}^{t-1}(1+\eta r_s^j)} \qquad \text{(ML-Prod)}$$

or, equivalently, $w_{1,k} = \frac{1}{K}$ and

$$w_{t+1,k} = \frac{w_{t,k}(1+\eta r_t^k)}{\sum_{j=1}^{K} w_{t,j}(1+\eta r_t^j)} \qquad \text{(ML-Prod, incremental)}$$

ML-Prod upper bound

Theorem: The regret of ML-Prod does not exceed $R_T \leq \frac{\ln K}{\eta} + \eta T$ for all $T \geq 0$.

Proof: Define potential function

$$\Phi_T = \sum_{k=1}^{K} \frac{1}{K} \prod_{t=1}^{T} (1 + \eta r_t^k)$$

Crucial observation is the *invariant* $\Phi_T = 1$. Clearly $\Phi_0 = 1$. And

$$
\begin{aligned}
&\Phi_{T+1} - \Phi_T \\
&= \sum_{k=1}^{K} \frac{1}{K} \prod_{t=1}^{T} (1 + \eta r_t^k) \eta r_{T+1}^k \\
&= \sum_{k=1}^{K} \frac{1}{K} \prod_{t=1}^{T} (1 + \eta r_t^k) \eta \left(\boldsymbol{w}_{T+1}^{\intercal} \boldsymbol{\ell}_{T+1} - \ell_{T+1}^k \right) \\
&= \sum_{k=1}^{K} \frac{1}{K} \prod_{t=1}^{T} (1 + \eta r_t^k) \left(\frac{\sum_j \prod_{s=1}^{t-1} (1 + \eta r_s^j) \eta \ell_{T+1}^j}{\sum_{j=1}^{K} \prod_{s=1}^{t-1} (1 + \eta r_s^j)} - \eta \ell_{T+1}^k \right) \\
&= \left(\sum_j \prod_{s=1}^{t-1} (1 + \eta r_s^j) \eta \ell_{T+1}^j - \sum_{k=1}^{K} \frac{1}{K} \prod_{t=1}^{T} (1 + \eta r_t^k) \eta \ell_{T+1}^k \right) \\
&= 0
\end{aligned}
$$

ML-Prod upper bound (ctd)

Since $\Phi_T = 1$, we find, for any k,

$$\prod_{t=1}^{T}(1+\eta r_t^k) \leq K$$

Taking logs, and using $\ln(1+\eta r) \geq \eta r - \eta^2$ for $r \geq -1$, we find

$$\sum_{t=1}^{T} \eta r_t^k - T\eta^2 \leq \sum_{t=1}^{T} \ln(1+\eta r_t^k) \leq \ln K$$

Reorganising, we find

$$\sum_{t=1}^{T} r_t^k \leq \frac{\ln K}{\eta} + T\eta$$

Specialists

Motivation

Not all expert predictions/actions available every round.

- Missing data
- Noise
- Too expensive ($/time/memory)

How to model missingness? Adversarial.

How to redefine the objective? New variant of regret.

How to still do something optimal? Upgrade of AA.

Mix loss game with specialists

Protocol:

- For $t = 1, 2, \ldots$
  - Adversary picks the subset $A_t \subseteq [K]$ of *awake* specialists.
  - Learner chooses a distribution $\boldsymbol{w}_t$ on awake specialists A_t.
  - Adversary reveals loss vector $\boldsymbol{\ell}_t \in (-\infty, \infty]^{A_t}$.
  - Learner's loss is the **mix loss** $-\ln\left(\sum_{k \in A_t} w_{t,k} e^{-\ell_{t,k}}\right)$

Objective

Loss *undefined* when a specialist is asleep.

Regret w.r.t. specialist j: only measured during rounds where j is awake

$$R_T^j = \sum_{\substack{t\in[T]\\ j\in A_t}} \underbrace{-\ln\left(\sum_{k\in A_t} w_t^k e^{-\ell_t^k}\right)}_{\text{Learner's mix loss in round } t} - \sum_{\substack{t\in[T]\\ j\in A_t}} \ell_t^j$$

Specialist AA

Definition: The *Specialist Aggregating Algorithm* (SAA) maintains a distribution $\boldsymbol{u}_t$. It starts uniform $\boldsymbol{u}_1^k = 1/K$.

In round t with awake experts A_t, SAA predict with

$$w_t^k \;=\; u_t(k|A_t) \;=\; \frac{u_t^k \mathbf{1}_{\{k\in A_t\}}}{\sum_{j\in A_t} u_t^j}$$

Update:

$$u_{t+1}^k \;=\; \begin{cases} \dfrac{u_t^k e^{-\ell_t^k}}{\sum_{j\in A_t} u_t^k e^{-\ell_t^k}} \sum_{j\in A_t} u_t^k & k \in A_t \\ u_t^k & k \notin A_t \end{cases}$$

AA update relative to awake set A_t

What makes this tick

Consider the sequence $\boldsymbol{\ell}'_1, \boldsymbol{\ell}'_2$ obtained by completing $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2$ by assigning in each round the SAA mix loss to all the asleep specialists.

Theorem: SAA on $\boldsymbol{\ell}$ and AA on $\boldsymbol{\ell}'$ produce identical weights $\boldsymbol{u}_t = \boldsymbol{w}'_t$ and suffer identical mix loss.

Proof technique for SAA

Proof:

Let u_t^k be the weights kept by SAA under loss ℓ and let ${w'_t}^k$ be the weights played by AA under loss ℓ'. We show that $u_t^k = {w'_t}^k$ by induction. By the algorithms' definition, we have $u_1^k = {w'_1}^k = \frac{1}{K}$. Let our inductive hypothesis be $u_t^k = {w'_t}^k$. This implies that $\sum_k u_t^k = 1$ (which can also be checked from the form of the u_{t+1}^k update). We define

$$
{\ell'}_t^k = \begin{cases} \ell_t^k, & \text{if } k \in A_t, \\ -\log \sum_{k \in A_t} w_t^k e^{-\ell_t^k} & \text{otherwise} \end{cases}
$$

where $w_t^k = \frac{u_t^k \mathbf{1}_{\{k \in A_t\}}}{\sum_{j \in A_t} u_t^j}$. The definition of ${\ell'}_t^k$ causes SAA and AA suffer the same loss (check that this is true for $k \in A_t$ and $k \notin A_t$).

We begin by calculating the normalization term of the ${w'}_t^k$ update:

$$\begin{aligned}
\sum_{k=1}^{K} {w'}_t^k e^{-{\ell'}_t^k} &= \sum_{k=1}^{K} u_t^k e^{-{\ell'}_t^k} \\
&= \sum_{k \in A_t} u_t^k e^{-\ell_t^k} + \sum_{k \notin A_t} u_t^k e^{\log \sum_{i \in A_t} w_t^i e^{-\ell_t^i}} \\
&= \sum_{k \in A_t} u_t^k e^{-\ell_t^k} + \sum_{i \in A_t} w_t^i e^{-\ell_t^i} \sum_{k \notin A_t} u_t^k \\
&= \sum_{k \in A_t} u_t^k e^{-\ell_t^k} + \sum_{k \in A_t} u_t^k e^{-\ell_t^k} \frac{\sum_{k \notin A_t} u_t^k}{\sum_{k \in A_T} u_t^k} \\
&= \sum_{k \in A_t} u_t^k e^{-\ell_t^k} \left(1 + \frac{\sum_{k \notin A_t} u_t^k}{\sum_{k \in A_T} u_t^k} \right) \\
&= \sum_{k \in A_t} u_t^k e^{-\ell_t^k} \frac{\sum_{k=1}^{K} u_t^k}{\sum_{k \in A_T} u_t^k} = \frac{\sum_{k \in A_t} u_t^k e^{-\ell_t^k}}{\sum_{k \in A_T} u_t^k}
\end{aligned}$$

It in now easy to show that ${w'}_t^k = u_t^k$. If $k \in A_t$, then

$$\begin{aligned}
{w'}_{t+1}^k &= \frac{{w'}_t^k e^{-{\ell'}_t^k}}{\sum_{k=1}^K {w'}_t^k e^{-{\ell'}_t^k}} \\
&= \frac{u_t^k e^{-\ell_t^k}}{\sum_{k \in A_t} u_t^k e^{-\ell_t^k}} \sum_{k \in A_T} u_t \\
&= u_{t+1}^k .
\end{aligned}$$

Otherwise, if $k \notin A_t$, then

$$\begin{aligned}
{w'}^k_{t+1} =& \frac{{w'}^k_t e^{-{\ell'}^k_t}}{\sum_{k=1}^K {w'}^k_t e^{-{\ell'}^k_t}} \\
=& \frac{u^k_t e^{\log \sum_{i \in A_t} w^i_t e^{-\ell^i_t}}}{\sum_{k \in A_t} u^k_t e^{-\ell^k_t}} \sum_{k \in A_T} u_t \\
=& u^k_t \frac{\sum_{k \in A_T} u_t}{\sum_{k \in A_T} u_t} \\
=& u^k_{t+1}.
\end{aligned}$$

Specialist regret bound for SAA

The AA has small regret w.r.t. expert j:

$$\begin{aligned}
\ln K &\geq \sum_{t=1}^{T} -\ln\left(\sum_{k=1}^{K} {w'_t}^k e^{-{\ell'_t}^k}\right) - \sum_{t=1}^{T} {\ell'_t}^j \\
&= \sum_{t=1}^{T} -\ln\left(\sum_{k\in A_t} w_t^k e^{-\ell_t^k}\right) - \sum_{\substack{t\in[T]\\ j\in A_t}} \ell_t^j - \sum_{\substack{t\in[T]\\ j\notin A_t}} -\ln\left(\sum_{k\in A_t} w_t^k e^{-\ell_t^k}\right) \\
&= \sum_{\substack{t\in[T]\\ j\in A_t}} -\ln\left(\sum_{k\in A_t} w_t^k e^{-\ell_t^k}\right) - \sum_{\substack{t\in[T]\\ j\in A_t}} \ell_t^j \\
&= R_T^j
\end{aligned}$$

Adversary more power (control over sleeping) but regret still $\ln K$: SAA minimax for specialist mix-loss regret game.

Specialists trick

Experts usually of our own design. They don't sleep.

However, *modelling* in terms of meta-specialists gives new insight/power

- Alarm-clocked experts: (k, s) wakes up in round s and predicts like k
- Subset experts: (k, S) predicts like k in rounds $t \in S \subseteq \{\text{time}\}$.
- Multitasking: (k, M) predicts like k on task $m_t \in M \subseteq \{\text{tasks}\}$.